
参赛密码 _____
(由组委会填写)

第十一届华为杯全国研究生数学建模竞赛

学 校 北京航空航天大学

参赛队号 10006002

队员姓名 1.赵礼兵

2.杜光勋

3.赵峙尧

参赛密码 _____
(由组委会填写)



第十一届华为杯全国研究生数学建模 竞赛

题 目 机动目标的跟踪与反跟踪 (B 题)

摘 要:

本论文针对机动目标的跟踪与反跟踪问题进行了详细地研究、分析和仿真验证。

针对问题 1，通过对该题的分析，我们将本问题分解为以下主要任务：(1) 建立对该目标的跟踪模型，并利用三个雷达测量数据在线实时估计目标的航迹变化；(2) 利用目标航迹跟踪结果，分析目标机动发生的时间范围，并统计目标加速度的大小和方向。针对任务 (1)，我们利用二阶常速 CV 模型对目标进行运动学建模，并为其设计卡尔曼状态估计器实时在线跟踪目标航迹。仿真结果显示，目标在 Z 轴方向的速度均值趋近于零，而在 X-Y 平面上的速度持续变化，并且在 X-Y 平面航迹主要包括水平定直前飞、水平转弯和水平盘旋。针对任务 (2)，我们利用航迹跟踪的结果，对三轴速度分别进行差分得到目标的三轴加速度变化情况，并利用目标 X 轴和 Y 轴加速度计算目标做水平转弯和盘旋时的向心加速度大小及其相对于 Y 轴正向的方位角变化。在此基础上分析了目标机动行为和发生的时间范围，目标航迹和加速度方向均以图形方式给出。

针对问题 2，通过该题的分析，我们将本题分解为以下主要任务：(1) 制定两个目标的航迹数据关联准则，对目标的实际验飞考核的飞行包线数据进行关联；(2) 设计实时在线航迹跟踪策略，跟踪两个目标的航迹；(3) 当出现雷达一段时间只有一个回波点迹的状况，设计一种方法使得航迹不丢失并给出处理结果。针对任务 (1)，我们采用二阶匀速 CV 模型对目标进行运动学建模，并采用简单实用的最近邻数据关联方法，通过设定每个目标的跟踪门限值来判断数据是否属于该目标。为了跟踪两个目标的航迹，我们针对任务 (2) 为两个目标分别设计了基于线性卡尔曼滤波器的航迹跟踪器，并实现了两个目标的序

贯实时航迹关联与跟踪。由于原始数据中两个目标的航迹存在交叉，并且每一时间段内只有一个回波点迹，所以我们针对任务（3）给出了一种限定目标航迹跟踪器状态更新周期以及放宽目标二跟踪门限的策略来实现目标航迹不丢失。仿真实现过程中我们发现，两个目标分别做交叉蛇形机动，从远距离点向雷达所在位置快速机动。

针对问题 3，通过对该题的分析，我们将本问题分解为以下主要任务：（1）构建新的目标跟踪模型，跟踪目标航迹；（2）根据目标加速度变化情况，分析目标的机动变化规律。我们首先利用了问题 1 中的二阶匀速（CV）模型对目标航迹进行跟踪。仿真结果显示目标跟踪效果不理想，分析发现目标速度持续变化，二阶匀速模型不再适用本题目标航迹跟踪的任务。因此，针对任务（1），我们利用三阶匀加速（CA）模型对目标进行运动学建模，并为其设计卡尔曼状态估计器实时在线跟踪目标航迹。仿真结果表明，该模型能够在线跟踪目标航迹，跟踪效果非常理想，该目标轨迹为一枚导弹自发射时刻起，在空中飞行的轨迹（并未着落）。针对任务（2），我们通过对航迹跟踪结果分析，我们发现目标在 X 轴和 Y 轴方向的速度基本不变，加速度均值趋近于零，在 Z 轴方向速度大小先减小后增大，加速度大小基本不变，约为重力加速度，方向竖直向下。因此，目标在发射升空后，处于近无动力飞行状态。

针对问题 4，通过对该题的分析，我们将本问题分解为以下主要任务：（1）构建目标跟踪模型，利用卡尔曼状态预测器实时预测目标轨迹；（2）根据问题 3 末段的目标轨迹估计结果，估计目标着落点坐标；（3）分析算法复杂度。针对任务（1），我们参考问题 3，建立目标的三阶匀加速（CA）模型，并利用卡尔曼状态预测器预测目标状态，实时预测目标轨迹。针对任务（2），我们利用任务（1）中航迹预测结果对目标轨迹进行迭代预测，进而估计着落点坐标。由仿真结果可知，目标着落点的经纬度坐标为 $(116.1738^{\circ}, 39.9585^{\circ})$ 。针对任务（3），

我们利用等效 flops 复杂度分析方法，对卡尔曼状态估计器和预测器的算法复杂度进行了分析。

针对问题 5，通过对该题的分析，我们将本问题分解为以下主要任务：（1）分析总结问题 2 中跟踪模型的不足；（2）根据雷达参数，确定雷达跟踪目标时的测量范围；（3）针对问题 2 中跟踪模型的不足之处，以及雷达跟踪目标时的测量范围，给出目标应该采用的逃逸策略与方案；（4）针对目标逃逸策略与方案，给出雷达跟踪策略调整方案。针对任务（1），我们对问题 2 的跟踪模型及策略进行了分析，发现二阶 CV 模型的特性，决定了不能很好的跟踪加速度随时间变化的机动目标，而且，跟踪策略假定只有两个目标需要跟踪，利用最近邻数据关联方法关联目标航迹，并假设所有数据为目标数据而非杂波。针对任务（2），通过对雷达参数的分析可知：如果目标 1 与雷达连线和目标 2 与雷达连线之间的夹角大于 3 度，则雷达不能同时锁定两个目标；目标距离雷达越近，目标距离雷达锁定范围边界的距离越小，也越容易逃脱锁定。针对任务（3），我们给出被雷达锁定的两个目标的逃逸策略：两目标分别释放两个干扰弹，同时以垂直于目标与雷达连线的速度方向，逆向分别进行变加速度机动，或者在三维空间中轨迹相互交叉，进行变加速机动。因此，在任务（4）中，为了保证对两个目标的跟踪，我们需要将跟踪模型改为交互多模型 IMM，并且将数据关联算法改为联合概率数据关联算法。

关键词：多目标跟踪，数据关联，卡尔曼估计与预测，算法复杂度

目 录

1	问题重述	1
1.1	问题背景.....	1
1.2	问题难点.....	1
1.3	主要研究问题.....	1
1.4	研究思路.....	2
2	符号说明	3
3	模型建立	4
3.1	目标运动状态方程.....	4
3.1.1	CV 模型	4
3.1.2	CA 模型	5
3.2	雷达测量方程.....	6
4	卡尔曼状态估计器与预测器	8
4.1	目标跟踪模型统一表述.....	8
4.2	卡尔曼状态估计器.....	8
4.3	卡尔曼状态预测器.....	9
5	问题 1：单目标航迹跟踪与机动检测	10
5.1	问题分析与假设.....	10
5.1.1	问题描述.....	10
5.1.2	数据初步分析.....	10
5.1.3	问题难点.....	10
5.1.4	问题假设.....	11
5.2	问题解决方案.....	11
5.2.1	解决思路.....	11
5.2.2	构建目标跟踪模型.....	12
5.2.3	目标航迹跟踪.....	14
5.2.4	机动行为检测与加速度信息统计.....	15
5.3	仿真结果与分析.....	16
5.4	小结.....	19
6	问题 2：多目标航迹跟踪与数据关联	21
6.1	问题分析与假设.....	21
6.1.1	问题描述.....	21
6.1.2	数据初步分析.....	21
6.1.3	问题难点.....	22
6.1.4	问题假设.....	22
6.2	问题解决方案.....	22
6.2.1	解决思路.....	22
6.2.2	基于最近邻近方法的概率数据关联算法.....	23
6.2.3	构建目标跟踪模型.....	24
6.2.4	目标数据关联与航迹跟踪.....	25
6.3	仿真结果与分析.....	27
6.4	小结.....	30
7	问题 3：加速度随时间变化目标的机动变化规律分析	31

7.1	问题分析与假设.....	31
7.1.1	问题描述.....	31
7.1.2	数据初步分析.....	31
7.1.3	问题难点.....	31
7.1.4	问题假设.....	31
7.2	问题解决方案.....	32
7.2.1	解决思路.....	32
7.2.2	构建目标跟踪模型.....	32
7.2.3	目标航迹跟踪.....	34
7.2.4	机动行为检测.....	34
7.3	仿真结果与分析.....	35
7.4	第 1 问中 CV 模型跟踪目标航迹结果分析	37
7.5	小结.....	38
8	问题 4: 目标轨迹实时预测与着落点估计	39
8.1	问题分析与假设.....	39
8.1.1	问题描述.....	39
8.1.2	数据初步分析.....	39
8.1.3	问题难点.....	39
8.1.4	问题假设.....	39
8.2	问题解决方案.....	40
8.2.1	解决思路.....	40
8.2.2	构建目标跟踪模型.....	40
8.2.3	目标轨迹实时预测与着落点估计.....	41
8.3	仿真结果与分析.....	42
8.4	算法复杂度计算分析.....	43
8.5	小结.....	47
9	问题 5: 目标的跟踪与反跟踪	48
9.1	问题分析.....	48
9.2	跟踪模型特性分析.....	48
9.3	雷达锁定范围分析.....	48
9.4	目标逃逸策略与方案.....	49
9.5	跟踪策略改进.....	49
9.6	小结.....	50
10	总结	51
	参考文献.....	53
	参赛体会.....	56
	附 录.....	57
	A.问题 1MATLAB 仿真程序	57
	B.问题 2MATLAB 仿真程序	63
	C.问题 3MATLAB 仿真程序	67
	D.问题 4MATLAB 仿真程序	74

1 问题重述

1.1 问题背景

目标跟踪是指根据传感器（如雷达等）所获得的对目标的测量信息，连续地对目标的运动状态进行估计，进而获取目标的运动态势及意图。目标跟踪理论在军、民用领域都有重要的应用价值。在军用领域，目标跟踪是情报搜集、战场监视、火力控制、态势估计和威胁评估的基础；在民用领域，目标跟踪被广泛应用于空中交通管制，目标导航以及机器人的道路规划等行业。

目标机动是指目标的速度大小和方向在短时间内发生变化，通常采用加速度作为衡量指标。目标机动与目标跟踪是“矛”与“盾”的关系。随着估计理论的日趋成熟及平台能力提升，目标作常规的匀速或者匀加速直线运动时的跟踪问题已经得到很好的解决。但被跟踪目标为了提高自身的生存能力，通常在被雷达锁定情况下会作规避的机动动作或者释放干扰力图摆脱跟踪，前者主要通过自身运动状态的快速变化导致雷达跟踪器精度变差甚至丢失跟踪目标，后者则通过制造假目标掩护自身，因此引入了在目标进行机动时雷达如何准确跟踪的问题。

1.2 问题难点

机动目标跟踪的难点在于以下几个方面：(1) 描述目标运动的模型[1][2]即目标的状态方程难于准确建立。通常情况下跟踪的目标都是非合作目标，目标的速度大小和方向如何变化难于准确描述；(2) 传感器自身测量精度有限加之外界干扰，传感器获得的测量信息[3]如距离、角度等包含一定的随机误差，用于描述传感器获得测量信息能力的测量方程难于完全准确反映真实目标的运动特征；(3) 当存在多个机动目标时，除了要解决(1)、(2)两个问题外，还需要解决测量信息属于哪个目标的问题，即数据关联。在一定的测量精度下，目标之间难于分辨，甚至当两个目标距离很近的时候，传感器往往只能获得一个目标的测量信息。由于以上多个挑战因素以及目标机动在战术上主动的优势，机动目标跟踪已成为近年来跟踪理论研究的热点和难点。

1.3 主要研究问题

问题 1：目标航迹估计与机动检测

根据附件中的 Data1.txt 数据，分析目标机动发生的时间范围，并统计目标加速度的大小和方向。建立对该目标的跟踪模型，并利用多个雷达的测量数据估计出目标的航迹。鼓励在线跟踪。

问题 2：不同目标的数据关联

附件中的 Data2.txt 数据对应两个目标的实际检飞考核的飞行包线（检飞：军队根据国家军标规则设定特定的飞行路线用于考核雷达的各项性能指标，因此包线是有实战意义的）。

本题最好利用序贯实时的方法完成各目标的数据关联，形成相应的航迹，并阐明所采用或制定的准则。若出现雷达一段时间只有一个回波点迹的状况，

应给出处理结果。

问题 3：加速度随时间变化目标的机动变化规律分析

根据附件中 Data3.txt 的数据，分析空间目标的机动变化规律(目标加速度随时间变化)。若采用第 1 问的跟踪模型进行处理，结果会有哪些变化。

问题 4：目标估计实时预测与着落点估计

根据第 3 问的目标轨迹对目标轨迹进行实时预测，同时估计该目标的着落点的坐标，给出详细结果，并分析算法复杂度。

问题 5：目标的跟踪与反跟踪

Data2.txt 数据中的两个目标已被雷达锁定跟踪。在目标能够及时了解是否被跟踪，并已知雷达的测量精度为雷达波束宽度为 3° ，即在以雷达为锥顶，雷达与目标连线为轴，半顶角为 1.5° 的圆锥内的目标均能被探测到；雷达前后两次扫描时间间隔最小为 0.5s。为应对你们的跟踪模型，目标应该采用怎样的有利于逃逸的策略与方案？反之为了保持对目标的跟踪，跟踪策略又应该如何相应地变换。

1.4 研究思路

首先，建立目标的跟踪模型。针对速度变化不大的目标，本论文给出了二阶匀速(CV)模型；针对加速运动，但是加速度变化不大的目标，本论文给出了三阶匀加速(CA)模型。由于雷达给出以雷达为原点的极坐标系下的数据，而所建立模型为直角坐标系，因此本论文根据坐标转换关系给出了观测量的随机统计特性。具体内容见论文第三章。

然后，根据所建立的目标跟踪模型，设计用于航迹估计的卡尔曼滤波器以及用于航迹预测的卡尔曼预测器，给出卡尔曼滤波与预测的递推方程，具体见论文第四章。

针对问题 1，利用 Data1.txt 中的数据以及二阶匀速(CV)跟踪模型，利用多个雷达给出的测量数据实时在线估计目标的航迹，并通过跟踪结果统计目标的加速度大小和方向，具体见论文第五章。

针对问题 2，本论文采用二阶匀速(CV)跟踪模型，利用最近邻数据关联方法以及限定目标航迹更新周期的方法，实现两个目标的航迹跟踪，具体见论文第六章。

针对问题 3，由于目标加速度随时间变化，因此采用三阶匀加速(CA)模型实现目标的航迹跟踪，具体见论文第七章。

针对问题 4，采用三阶匀加速(CA)，为预测目标航迹设计了卡尔曼预测器，实时预测出目标的航迹以及可能的着落点。最后分析了算法的时间复杂度和空间复杂度，具体见论文第八章。

针对问题 5，本论文对跟踪 Data2.txt 中两个目标时采用的跟踪模型进行分析，给出目标应该采用的逃逸策略和方案，为了保持对目标的跟踪，本论文给出了相应的跟踪策略，具体见论文第九章。

最后，第十章对本论文进行了总结。

2 符号说明

序号	符号	符号说明
1	r	极坐标系下目标距离
2	φ	极坐标系下目标方位角
3	θ	极坐标系下目标俯仰角
4	x	直角坐标系下目标在 x 方向的坐标
5	y	直角坐标系下目标在 y 方向的坐标
6	z	直角坐标系下目标在 z 方向的坐标
7	v_x	直角坐标系下目标在 x 方向的速度
8	v_y	直角坐标系下目标在 y 方向的速度
9	v_z	直角坐标系下目标在 z 方向的速度
10	a_x	直角坐标系下目标在 x 方向的加速度
11	a_y	直角坐标系下目标在 y 方向的加速度
12	a_z	直角坐标系下目标在 z 方向的加速度
13	g	重力加速度，取值 9.8m/s^2

3 模型建立

本论文研究的是一个三维空间的雷达，雷达可以探测到目标的径向距离、方位角和俯仰角。一般来说，如果目标做匀速直线运动时，跟踪问题十分容易，但当目标做机动时，由于无法准确预知目标下一时刻的运动状态，使得跟踪变得很困难，这就需要采用合适的目标运动模型，现在的各种模型大致分为单模型和多模型，由于多模型较为复杂，这里我们采用单模型方法。常用的单模型[1],[2]有匀速模型(CV)、匀加速模型(CA)、Signer 模型和均值自适应的“当前”统计模型(CS)；多模型有交互式多模型(IMM)。本论文利用二阶匀速模型(CV)、三阶匀加速模型(CA)对目标航迹进行跟踪[1]-[21]。

3.1 目标运动状态方程

3.1.1 CV 模型

如果目标处于匀速直线运动，或者速度变化不大的时候，它的状态可以用离散匀速模型来表述。考虑到目标的运动状态往往受到一定程度的随机噪声干扰，或者目标的速度会进行小幅度的变化，故我们假设目标各个方向的速度变化看作是具有随机扰动特性的扰动输入，并假设其服从零均值白色高斯分布，即

$$\begin{cases} \dot{v}_x = 0 + w_1 \\ \dot{v}_y = 0 + w_2 \\ \dot{v}_z = 0 + w_3 \end{cases} \quad (3.1)$$

其中， $w = [w_1, w_2, w_3]^T$ 为均值为零、方差已知的高斯分布。此时，目标作近匀速运动，故可采用二阶匀速（CV）模型[1],[2]对目标建模。

对于目标运动学方程，我们有：

$$X(k+1) = \Phi X(k) + \Gamma w(k) \quad (3.2)$$

其中 $X = [x, y, z, v_x, v_y, v_z]^T$ ，状态转移矩阵 Φ 和系统噪声驱动矩阵 Γ 分别为：

$$\Phi = \begin{bmatrix} I_3 & TI_3 \\ 0 & I_3 \end{bmatrix}, \Gamma = \begin{bmatrix} \frac{1}{2}T^2I_3 \\ TI_3 \end{bmatrix} \quad (3.3)$$

其中 T 为测量数据更新周期， $w(k)$ 满足如下正态分布：

$$w \sim N(0, Q) \quad (3.4)$$

$$Q = E[\Gamma w w^T \Gamma^T] = \Gamma \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \Gamma^T \quad (3.5)$$

其中 Q 为系统噪声协方差阵。在二阶匀速 CV 模型下， $\sigma = [\sigma_1, \sigma_2, \sigma_3]^T$ 满足

$$\begin{cases} \sigma_1 \in [0.5a_{x_max}, a_{x_max}] \\ \sigma_2 \in [0.5a_{y_max}, a_{y_max}] \\ \sigma_3 \in [0.5a_{z_max}, a_{z_max}] \end{cases} \quad (3.6)$$

其中, $a_{x_max}, a_{y_max}, a_{z_max}$ 为各方向的最大加速度值, 具体数值可在目标跟踪时根据实际效果调节, 从而适应速度变化程度不同的目标。

3.1.2 CA 模型

如果目标处于匀加速运动, 或者加速度变化不大的时候, 它的状态可以用离散匀加速模型来表述。考虑到目标的运动状态往往受到一定程度的随机噪声干扰, 或者目标的加速度会进行小幅度的变化, 故我们假设目标各个方向的加速度变化是具有随机扰动特性的扰动输入, 并假设其服从零均值白色高斯分布, 即

$$\begin{cases} \dot{a}_x = 0 + w_1 \\ \dot{a}_y = 0 + w_2 \\ \dot{a}_z = 0 + w_3 \end{cases} \quad (3.7)$$

其中, $w = [w_1, w_2, w_3]^T$ 为均值为零、方差已知的高斯分布。此时, 目标作近匀加速运动, 故可采用三阶匀加速 (CA) 模型[1]对目标建模。

对于目标运动学方程, 我们有:

$$X(k+1) = \Phi X(k) + \Gamma w(k) \quad (3.8)$$

其中 $X = [x, y, z, v_x, v_y, v_z, a_x, a_y, a_z]^T$, 状态转移矩阵 Φ 和系统噪声驱动矩阵 Γ 分别为:

$$\Phi = \begin{bmatrix} I_3 & TI_3 & 1/2T^2I_3 \\ 0 & I_3 & TI_3 \\ 0 & 0 & I_3 \end{bmatrix}, \Gamma = \begin{bmatrix} 1/2T^2I_3 \\ TI_3 \\ I_3 \end{bmatrix} \quad (3.9)$$

其中 T 为测量数据更新周期, $w(k)$ 满足如下正态分布:

$$w \sim N(0, Q) \quad (3.10)$$

$$Q = E[\Gamma w w^T \Gamma^T] = \Gamma \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \Gamma^T \quad (3.11)$$

其中 Q 为系统噪声协方差阵。在三阶匀加速模型(CA)下, $\sigma = [\sigma_1, \sigma_2, \sigma_3]^T$ 满足

$$\begin{cases} \sigma_1 \in [0.5\Delta a_x, \Delta a_x] \\ \sigma_2 \in [0.5\Delta a_y, \Delta a_y] \\ \sigma_3 \in [0.5\Delta a_z, \Delta a_z] \end{cases} \quad (3.12)$$

其中， $\Delta a_x, \Delta a_y, \Delta a_z$ 为各方向的加速度增量的最大幅值，具体数值可在目标跟踪时根据实际效果调节，从而适应加速度变化程度不同的目标。

3.2 雷达测量方程

因为目标运动在直角坐标系下最容易描述，但是实际上雷达给出的数据是极坐标系内的目标距离 r ，单位 (m)，目标方位角 φ ，单位 ($^\circ$)，目标俯仰角 θ ，单位 ($^\circ$)，绝对测量时间，单位 (s) 和对应传感器的标号，其中，数据测量的坐标系有如下规定：原点 O 为传感器中心，传感器中心点与当地纬度切线方向指向东为 x 轴，传感器中心点与当地经度切线方向指向北为 y 轴，地心与传感器中心连线指向天向的为 z 轴，目标方位角指北向顺时针夹角（从 y 轴正向向 x 轴正向的夹角，范围为 $0 \sim 360^\circ$ ），目标俯仰角指传感器中心点与目标连线和地平面的夹角（即与 xOy 平面的夹角，通常范围 -90° 到 90° ），如图 2-1 所示。

令 $z_s = [r, \varphi, \theta]^T$ ， $Z = [x, y, z]^T$ ，则极坐标向直角坐标转换方程为

$$Z = \begin{bmatrix} x \\ y \\ z \end{bmatrix} = h(z_s) = \begin{bmatrix} r \cos \theta \sin \varphi \\ r \cos \theta \cos \varphi \\ r \sin \theta \end{bmatrix} \quad (3.13)$$

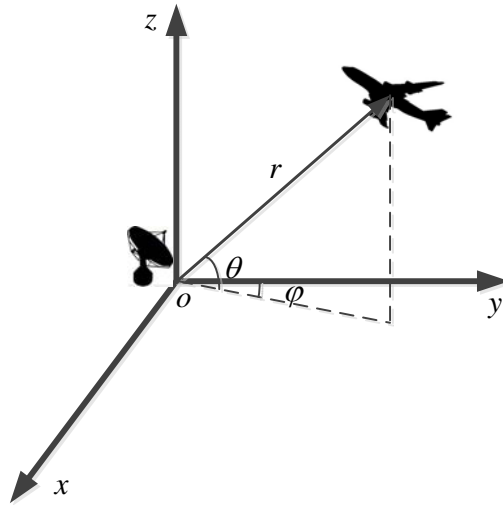


图 2-3-1 雷达测量示意图

直角坐标系内的测量方程为

$$Z(k) = HX(k) + v(k) \quad (3.14)$$

对于二阶 CV 模型,

$$H = [I_3 \quad 0_{3 \times 3}] \quad (3.15)$$

对于三阶 CA 模型,

$$H = [I_3 \quad 0_{3 \times 3} \quad 0_{3 \times 3}] \quad (3.16)$$

系统噪声 $v(k)$ 满足如下正态分布:

$$v(k) \sim N(0, R_L(k)) \quad (3.17)$$

其中 $R_L(k)$ 为直角坐标系下的测量噪声协方差阵, 满足[3]:

$$R_L(k) = J(z_s(k)) R J(z_s(k))^T \quad (3.18)$$

其中 Jacobian 矩阵 $J(z_s(k))$ 由实时的极坐标下测量值 $z_s(k)$ 确定:

$$\begin{aligned} J(z_s(k)) &= \left. \frac{\partial h(z_s)}{\partial z_s} \right|_{z_s=z_s(k)} \\ &= \begin{bmatrix} \cos \theta \sin \varphi & r \cos \theta \cos \varphi & -r \sin \theta \sin \varphi \\ \cos \theta \cos \varphi & -r \cos \theta \sin \varphi & -r \sin \theta \cos \varphi \\ \sin \theta & 0 & r \cos \theta \end{bmatrix} \end{aligned} \quad (3.19)$$

极坐标下的测量噪声协方差阵 R 由各个雷达的测量误差确定。

注意: R_L 为时变矩阵, 其值取决于极坐标系下当前时刻的距离、方位角和俯仰角的原始测量数据。而且, 针对不同的传感器, 极坐标系下的测量噪声协方差阵 R 不同。

由于雷达的测量误差已知分别为 $\Delta r, \Delta \varphi, \Delta \theta$ (角度单位转换为弧度), 本文对测量误差的理解为: 测量值与真实值最大偏差。假设测量误差服从正态分布, 根据 3σ 准则, 误差以 99.7% 的概率落在 $[-3\sigma, 3\sigma]$ 之间, 则我们近似认为

$$\sigma_r = \Delta r/3, \sigma_\varphi = \Delta \varphi/3, \sigma_\theta = \Delta \theta/3 \quad (3.20)$$

则测量噪声协方差阵 R 由下式给出

$$R = \begin{bmatrix} \sigma_r^2 & 0 & 0 \\ 0 & \sigma_\varphi^2 & 0 \\ 0 & 0 & \sigma_\theta^2 \end{bmatrix} \quad (3.21)$$

4 卡尔曼状态估计器与预测器

本论文在解决问题时，采用的目标跟踪模型均为线性模型，因此可以设计线性卡尔曼滤波器来跟踪目标航迹，设计线性卡尔曼预测器来预测目标航迹。本部分首先给出目标跟踪模型的统一表述，然后为其设计相应的线性卡尔曼滤波器和预测器[4],[7],[9],[11],[21],[31]-[33]。

4.1 目标跟踪模型统一表述

根据第三部分中给出的 CV 模型和 CA 模型，以及雷达测量方程可以将目标跟踪模型统一表述如下

$$\begin{aligned} X(k+1) &= \Phi X(k) + \Gamma w(k) \\ Z(k) &= HX(k) + v(k) \\ E[\Gamma w(k) w^T(k) \Gamma^T] &= Q \\ E[v(k) v^T(k)] &= R_L(k) \end{aligned} \quad (4.1)$$

4.2 卡尔曼状态估计器

根据文献[4]，为(4.1)中所示模型设计卡尔曼状态估计器如下：
计算状态估计值：

$$\hat{X}(k+1|k+1) = \hat{X}(k+1|k) + K(k+1)\tilde{z}(k+1) \quad (4.2)$$

计算状态一步预测：

$$\hat{X}(k+1|k) = \Phi \hat{X}(k|k) \quad (4.3)$$

计算新息：

$$\tilde{z}(k+1) = Z(k+1) - H\hat{X}(k+1|k) \quad (4.4)$$

计算卡尔曼滤波增益：

$$K(k+1) = P(k+1|k)H^T[HP(k+1|k)H^T + R_L]^{-1} \quad (4.5)$$

计算估计误差协方差矩阵：

$$P(k+1|k) = \Phi P(k|k)\Phi^T + Q \quad (4.6)$$

计算估计误差协方差矩阵：

$$P(k+1|k+1) = [I - K(k+1)H]P(k+1|k) \quad (4.7)$$

4.3 卡尔曼状态预测器

根据文献[4]，为(4.1)中所示模型设计卡尔曼状态预测器如下：

(1) 在 $k=0$ 时刻给定状态变量初值 $\hat{X}(0)=X(0)$ 和初始误差方差阵 $P(0)$ 。

(2) 计算 $k=0$ 时刻最优增益矩阵：

$$K(0)=\Phi P(0)H^T[HP(0)H^T+R_L]^{-1} \quad (4.8)$$

(3) 计算 $X(1)$ 的最优估值 $\hat{X}(1|0)$ ：

$$\hat{X}(1|0)=\Phi\hat{X}(0)+K(0)[Y(0)-H\hat{X}(0)] \quad (4.9)$$

(4) 更新误差方差阵：

$$P(1|0)=\Phi P(0)\Phi^T-\Phi P(0)H^T[HP(0)H^T+R_L]^{-1}HP(0)\Phi^T+Q \quad (4.10)$$

(5) 当 $k>0$ 时，卡尔曼预测的计算流程为：

$$\begin{cases} K(k)=\Phi P(k|k-1)H^T[HP(k|k-1)H^T+R_L]^{-1} \\ \hat{X}(k+1|k)=\Phi\hat{X}(k|k-1)+K(k)[Z(k)-H\hat{X}(k|k-1)] \\ P(k+1|k)=\Phi P(k|k-1)\Phi^T-\Phi P(k|k-1)H^T \\ \quad \cdot [HP(k|k-1)H^T+R_L]^{-1}HP(k|k-1)\Phi^T+Q \end{cases} \quad (4.11)$$

5 问题 1：单目标航迹跟踪与机动检测

5.1 问题分析与假设

5.1.1 问题描述

问题 1 表述如下：根据附件中的 Data1.txt 数据，分析目标机动发生的时间范围，并统计目标加速度的大小和方向。建立对该目标的跟踪模型，并利用多个雷达的测量数据估计出目标的航迹。鼓励在线跟踪。

在本问题中，三个雷达的位置坐标和测量误差如下：

表 5-1 雷达位置坐标和测量误差表

雷达标号	经度($^{\circ}$)	纬度($^{\circ}$)	高度(m)	测距误差(m)	方位角误差($^{\circ}$)	俯仰角误差($^{\circ}$)
1	122.1	40.5	0	50	0.4	0.4
2	122.4	41.5	0	40	0.3	0.3
3	122.7	40.9	0	60	0.5	0.5

通过分析可知，本题研究的主要任务如下：

- (1) 建立对目标的跟踪模型，并利用三个雷达测量数据在线实时估计目标的航迹变化。
- (2) 利用目标航迹跟踪结果，分析目标机动发生的时间范围，并统计目标加速度的大小和方向。

5.1.2 数据初步分析

利用 MATLAB 对 Data1.txt 进行数据提取，可知数据为一个 728×5 的数据矩阵。其中，前三列分别为目标的极坐标数据，第四列为数据测量的绝对时间，第五列为传感器编号。对第四列进行差分，可得到目标从第 0 秒到第 749 秒的测量数据，采样周期为 1 秒。

经过对数据分析，发现 Data1.txt 数据存在三个问题：

- 1、同一传感器的测量数据并不是均匀采样的；
- 2、传感器 1 和传感器 2 数据之间存在 45 秒的数据空白；
- 3、传感器 2 和传感器 3 在同一测量时间下均存在测量数据。

5.1.3 问题难点

通过对问题的初步分析，发现该问题的研究存在如下难点：

- (1) 由于未知目标的运动特性，因此无法确定哪种模型更适合用于目标航迹跟踪。
- (2) 雷达测量数据存在数据丢失情况，以及不同雷达的测量数据存在重叠。
- (3) 本题数据为三个传感器测量数据，数据的坐标系原点存在偏移。各传感器测量数据的测量误差不同。

(4) 如何制定机动动作判定准则, 辨识目标机动动作, 并确立机动发生的时间范围。

5.1.4 问题假设

假设 1: 在本题中, 将目标各个方向的速度变化看作是具有随机扰动特性的扰动输入, 并假设其服从零均值白色高斯分布。

基于假设 1, 我们可利用 CV 模型来跟踪目标航迹。

假设 2: 表 5-1 给出的雷达测量误差为测量值与真实值的最大偏差。

5.2 问题解决方案

5.2.1 解决思路

针对问题分析中本题目的两个主要任务, 解决方案如下:

首先, 根据假设 1 和假设 2, 建立目标的运动学方程模型, 其中状态变量为机动目标在直角坐标系下的位置和速度。

其次, 利用卡尔曼滤波方法估计模型状态变量, 获取目标航迹变化和目標速度变化。其中由于三个雷达原点位置不统一, 且数据为极坐标形式, 在进行卡尔曼状态估计过程中, 需要将雷达测量数据转换到以雷达 1 位置为原点的直角坐标系内。

第三, 根据目标速度的变化, 通过对速度估计结果差分, 得到目标加速度变化的方向和大小。

最后, 综合分析目标的位置、速度以及加速度变化情况, 判定目标机动行为及发生的时间范围。

研究思路如下图所示:

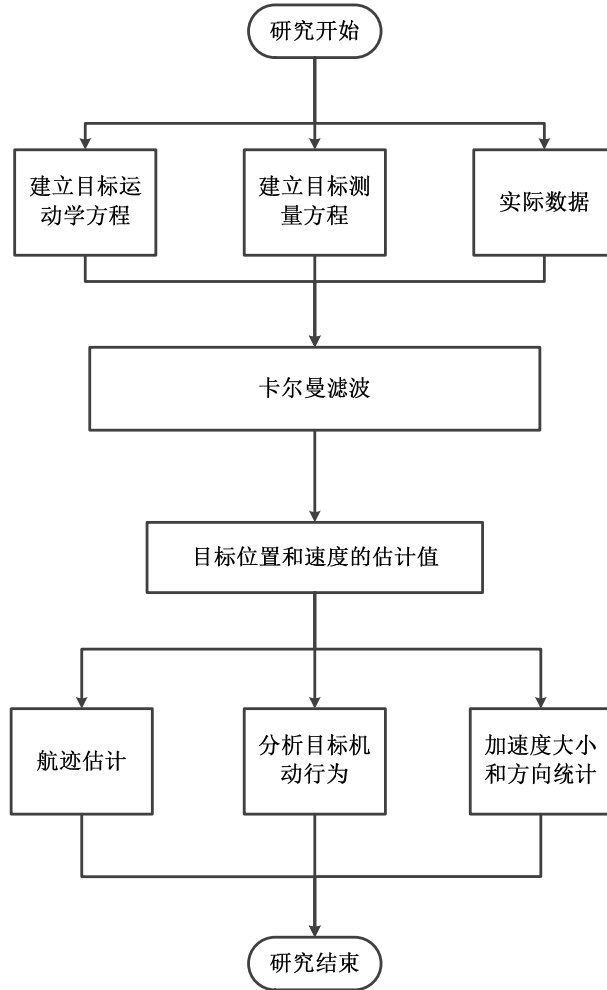


图 5-1 问题 1 研究思路

5.2.2 构建目标跟踪模型

根据本论文第二部分，我们得到目标跟踪 CV 模型如下：

$$X(k+1) = \Phi X(k) + \Gamma w(k) \quad (5.1)$$

其中 $X = [x, y, z, v_x, v_y, v_z]^T$ ，状态转移矩阵 Φ 和系统噪声驱动矩阵 Γ 分别为：

$$\Phi = \begin{bmatrix} I_3 & TI_3 \\ 0 & I_3 \end{bmatrix}, \Gamma = \begin{bmatrix} \frac{1}{2}T^2 I_3 \\ TI_3 \end{bmatrix} \quad (5.2)$$

其中 $T=1$ ， $w(k)$ 满足如下正态分布：

$$w \sim N(0, Q) \quad (5.3)$$

$$Q = E[ww^T] = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \quad (5.4)$$

其中 $\sigma = [\sigma_1, \sigma_2, \sigma_3]^T$,

$$\sigma_1 = 0.5a_{x_max}, \sigma_2 = 0.5a_{y_max}, \sigma_3 = 0.5a_{z_max} \quad (5.5)$$

其中, $a_{x_max}, a_{y_max}, a_{z_max}$ 为各方向的最大加速度值, 具体数值可在目标跟踪时根据实际效果调节, 从而适应速度变化程度不同的目标
直角坐标系内的测量方程为

$$Z(k) = HX(k) + v(k) \quad (5.6)$$

其中 $H = [I_3 \quad 0_{3 \times 3}]$, 系统噪声 $v(k)$ 满足如下正态分布:

$$v(k) \sim N(0, R_L(k)) \quad (5.7)$$

其中 $R_L(k)$ 为直角坐标系下的测量噪声协方差阵, 满足[3]:

$$R_L(k) = J(z_s(k)) R J(z_s(k))^T \quad (5.8)$$

其中 Jacobian 矩阵 $J(z_s(k))$ 由实时的极坐标下测量值 $z_s(k)$ 确定:

$$J(z_s(k)) = \left. \frac{\partial h(z_s)}{\partial z_s} \right|_{z_s = z_s(k)} = \begin{bmatrix} \cos \theta \sin \varphi & r \cos \theta \cos \varphi & -r \sin \theta \sin \varphi \\ \cos \theta \cos \varphi & -r \cos \theta \sin \varphi & -r \sin \theta \cos \varphi \\ \sin \theta & 0 & r \cos \theta \end{bmatrix} \quad (5.9)$$

极坐标下的测量噪声协方差阵 R 由各个雷达的测量误差确定。
基于假设 2, 雷达 1 在极坐标系下的测量误差协方差阵为

$$R = R_1 = \begin{bmatrix} (50/3)^2 & 0 & 0 \\ 0 & (0.4/3)^2 & 0 \\ 0 & 0 & (0.4/3)^2 \end{bmatrix} \quad (5.10)$$

同理, 对于雷达 2 和 3, 我们分别有:

$$R = R_2 = \begin{bmatrix} (40/3)^2 & 0 & 0 \\ 0 & (0.3/3)^2 & 0 \\ 0 & 0 & (0.3/3)^2 \end{bmatrix} \quad (5.11)$$

$$R = R_3 = \begin{bmatrix} (60/3)^2 & 0 & 0 \\ 0 & (0.5/3)^2 & 0 \\ 0 & 0 & (0.5/3)^2 \end{bmatrix} \quad (5.12)$$

5.2.3 目标航迹跟踪

利用 4.2 节给出的卡尔曼状态估计器，设计目标航迹跟踪算法，并对目标航迹进行跟踪。

考虑到雷达数据存在丢失、重叠等问题，我们不能直接利用第 4.2 节给出的卡尔曼滤波状态估计器进行目标跟踪，我们给出了修正的卡尔曼状态估计过程来跟踪目标航迹，算法步骤如下图所示。

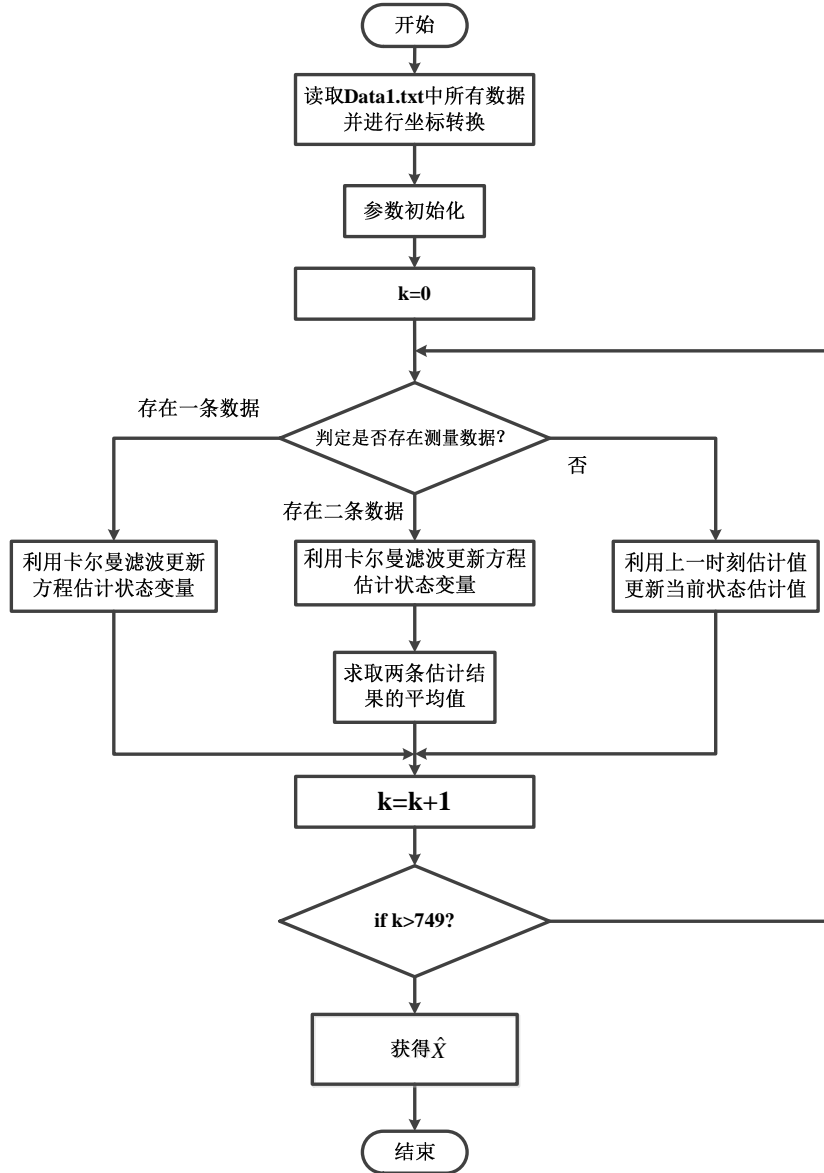


图 5-2 改进的基于卡尔曼滤波的目标估计算法流程图

(1) 读取 Data1.txt 中的所有测量数据，并将极坐标形式的目标测量值转换到以雷达 1 位置为原点的直角坐标系，转换公式如下

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r \cos \theta \sin \varphi \\ r \cos \theta \cos \varphi \\ r \sin \theta \end{bmatrix} + \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix} \quad (5.13)$$

其中 (x_i, y_i, z_i) 为雷达 i 在雷达 1 位置为原点的直角坐标系内的坐标。已知雷达 1 的经纬度为 $(122.1^\circ, 40.5^\circ)$ ，雷达 2 相对于雷达 1 的经纬度偏差为 $(0.3^\circ, 1^\circ)$ ，雷达 3 相对于雷达 1 的经纬度偏差为 $(0.6^\circ, 0.4^\circ)$ 。则

$$\begin{aligned}(x_1, y_1, z_1) &= (0, 0, 0) \\(x_2, y_2, z_2) &= \left(\frac{0.3\pi}{180} R_{earth} \cos\left(\frac{40.5\pi}{180}\right), \frac{\pi}{180} R_{earth}, 0 \right) \\(x_3, y_3, z_3) &= \left(\frac{0.6\pi}{180} R_{earth} \cos\left(\frac{40.5\pi}{180}\right), \frac{0.4\pi}{180} R_{earth}, 0 \right)\end{aligned}\quad (5.14)$$

其中 R_{earth} 为地球半径，取值为 6400km。

(2) $k = 0$

(3) 对于时刻 k ，检查是否有雷达返回测量数据（即查询 Data1.txt 中对应时间点是否有数据）。若存在 1 条测量数据，则跳转到步骤 (2)；若存在 2 条数据，则跳转到步骤 (3)；若不存在数据，则跳转到步骤 (4)。

(4) 此时，雷达返回 1 条测量数据，即传感器 1、2、3 有且仅有一个传感器返回了目标的测量数据。此时，直接利用卡尔曼状态估计器完成系统状态变量的估计。

(5) 此时，雷达返回 2 条测量数据，即传感器 2 和传感器 3 同时返回了目标的测量数据。为了减少误差对估计的影响，利用卡尔曼状态估计器，分别基于传感器 2 和传感器 3 数据估计出系统状态变量，再对估计结果平均化，实现数据融合[17]。

(6) 此时，雷达没有返回数据。这时，我们无法利用当前测试数据对状态变量进行更新。为了保证航迹估计的连续性，我们仅利用上一时刻的估计值对当前状态进行更新，更新公式如下

$$\begin{aligned}\hat{X}(k+1|k+1) &= \Phi \hat{X}(k|k) \\P(k+1|k) &= \Phi P(k|k) \Phi^T + Q \\K(k+1) &= P(k+1|k) H^T [H P(k+1|k) H^T + R_L]^{-1} \\P(k+1|k+1) &= [I - K(k+1) H] P(k+1|k)\end{aligned}\quad (5.15)$$

(7) $k = k + 1$ 。

(8) 若 $k > 749$ ，结束滤波；否则，重复 (3) - (7)。

值得一提的是，在第 (6) 步，我们只对采样间隔不均的情况进行基于卡尔曼滤波的短期预测（一般是 1 步或 2 步）。对于传感器 1 和传感器 2 测量数据之间约 45 秒的数据空白，考虑到长期预测的不确定性，本算法对该段时间不做预测。

注意：本算法，仅利用当前时刻数据以及历史数据实现了目标的位置估计和速度估计，是一种在线跟踪算法。

5.2.4 机动行为检测与加速度信息统计

机动动作判定准则：1) 加速度大小明显大于 0，方向持续改变，则机动作

作为水平转弯；2) 加速度大小明显大于 0，方向近似周期性改变，则机动动作作为水平盘旋。

通过以上目标航迹跟踪算法，可获得向量 x 的估计值。取向量 \hat{x} 中的 $[\hat{x}, \hat{y}, \hat{z}]$ 可获取目标的航迹；取向量 x 中的 $[\hat{v}_x, \hat{v}_y, \hat{v}_z]$ ，并对其进行差分，可获取目标的方向加速度的变化，在此基础上可统计出加速度的大小和方向，并分析出目标机动行为和发生的时间范围。

5.3 仿真结果与分析

运行 “\B10006002fj\MATLAB 仿真程序\问题 1\Main.m”，可以得出目标位置和速度的变化的估计值。

在这里，我们取各个方向的加速度最大值为 $1g$ 。基于此，我们在仿真中取

$$\sigma = [0.5g, 0.5g, 0.5g]^T \quad (5.16)$$

图 5-3 给出了目标的三维航迹图，其中红色星点为原始数据，蓝色圆点为航迹估计数据。为了方便读者观察其航迹，图 5-4 给出了目标航迹在 X-Y 平面上的投影。

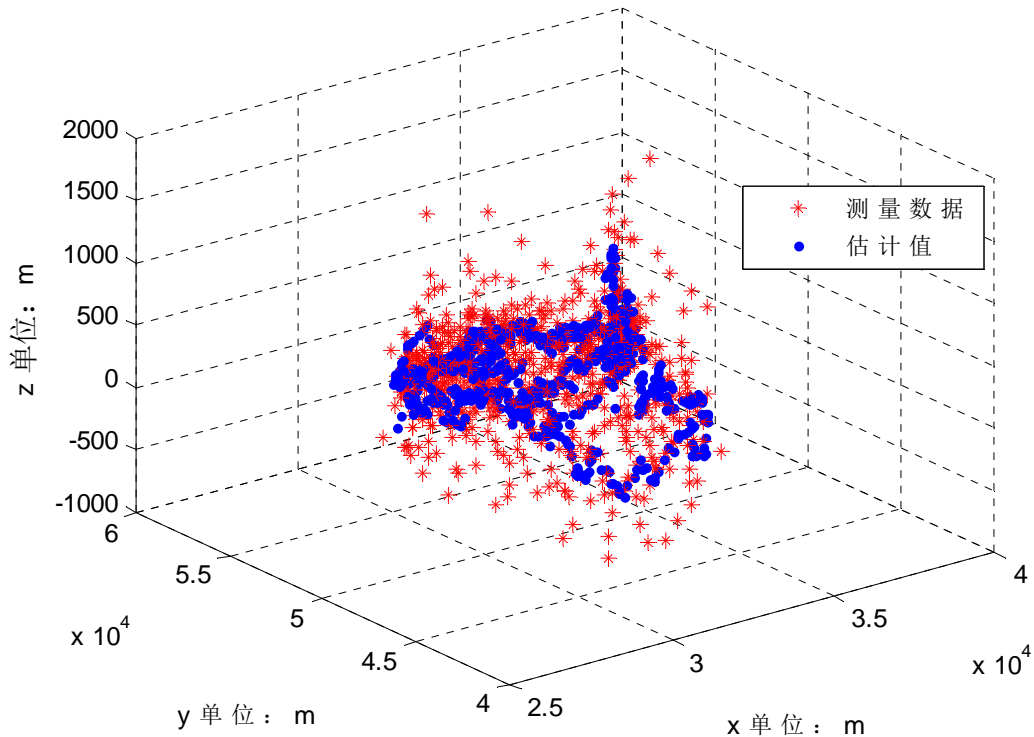


图 5-3 目标的三维航迹图

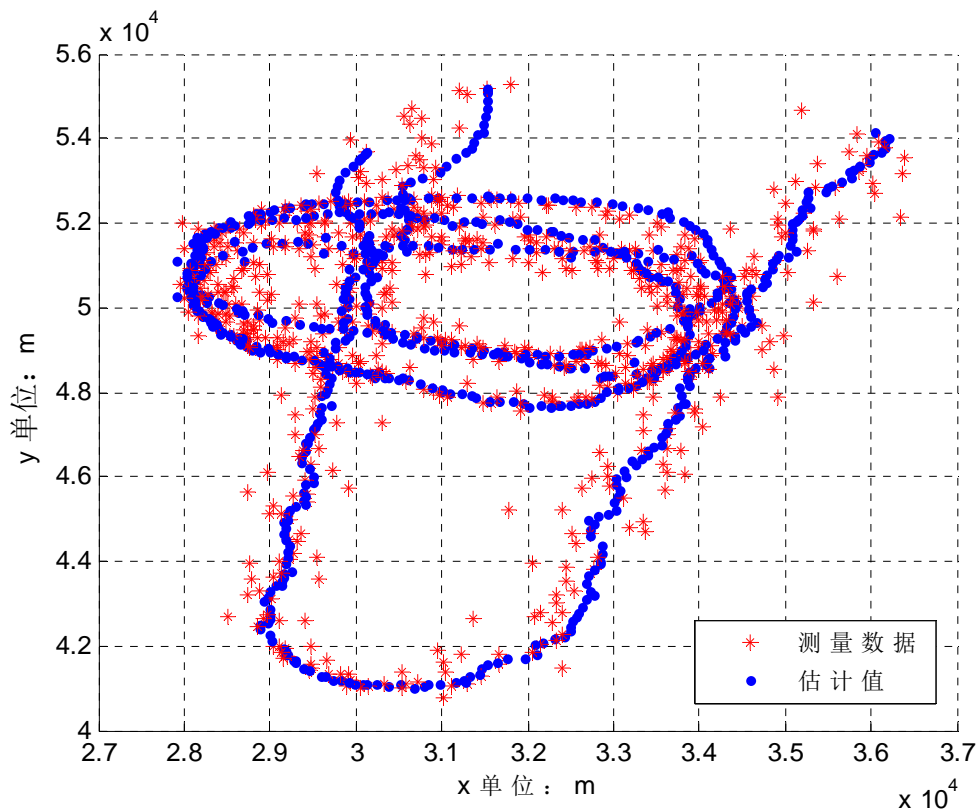


图 5-4 X-Y 平面的航迹视角图

图 5-5 给出了目标在各方向的速度变化图。

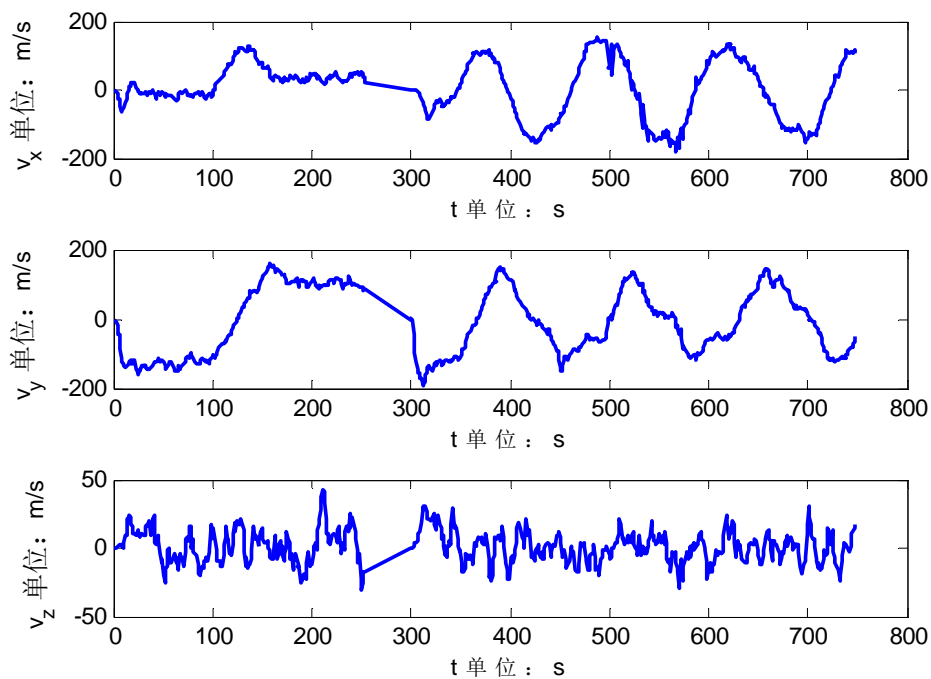


图 5-5 目标在各方向的速度变化图

通过对 v_x , v_y 和 v_z 进行差分, 可获取加速度在各方向的变化, 如图 5-6 所示。

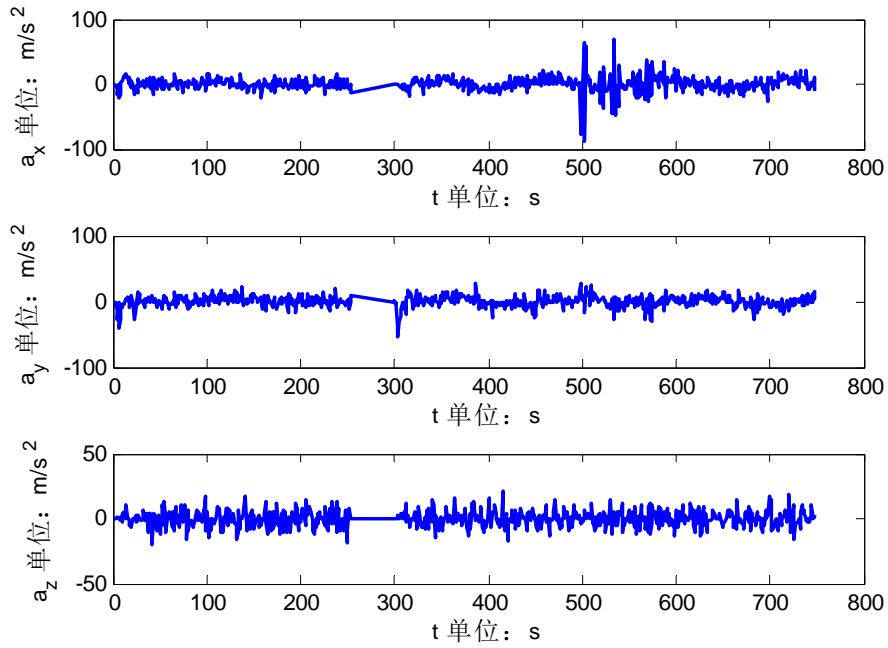


图 5-6 目标在各方向的加速度变化图

根据图 5-4, 在本题中我们更关心的 X-Y 平面上的向心加速度大小和方向, 我们有

$$\begin{cases} \|a\| = \sqrt{a_x^2 + a_y^2} \\ \varphi_a = \arctan \frac{a_x}{a_y} \end{cases} \quad (5.17)$$

图 5-7 给出了目标每个时刻点在 X-Y 平面上的向心加速度大小和方向变化图。

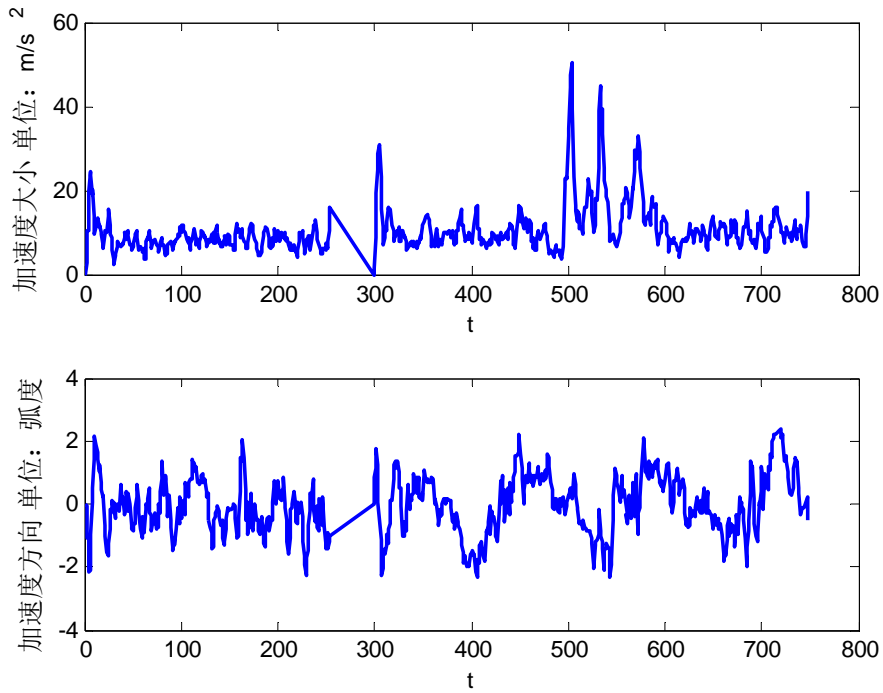


图 5-7 向心加速度大小和方向变化图

结合图 5-4 至图 5-7，目标的完整动作可分解为：

- (1) 沿 y 轴负向这个大方向做匀速直线运动；
- (2) 完成一个约逆时针 180 度的转弯，完成**第一次机动动作**；
- (3) 沿 y 轴正向这个大方向做匀速直线运动；
- (4) 完成一个约逆时针 90 度的转弯，完成**第二次机动动作**（无测量数据）；
- (5) 沿 x 轴负向这个大方向做匀速直线运动；
- (6) 完成一个约逆时针 90 度的转弯，完成**第三次机动动作**；
- (7) 沿 y 轴负向这个大方向做匀速直线运动；
- (8) 逆时针盘旋约三周，完成**第四次机动动作**；

通过对比图 5-5 至图 5-7，分析出机动动作的时间范围如下：

(1) 第一次机动动作（完成一个约逆时针 180 度的转弯）的时间范围大致在相对时间第 100 秒到第 160 秒（绝对时间第 3.6720400e+004 秒到第 3.6780400e+004 秒）左右。

(2) 第三次机动动作（完成两个逆时针 90 度的转弯）的时间范围大致在相对时间第 300 秒到第 320 秒（绝对时间第 3.6920400e+004 秒到第 3.6940400e+004 秒）左右。

(3) 第四次机动动作（逆时针盘旋两周）的时间范围大致在相对时间第 350 秒到第 749 秒（绝对时间第 3.7250400e+004 秒到第 3.7369400e+004 秒）左右。

在这里我们对第一、三和四次机动动作对应的加速度大小给出统计结果，如下表所示。

机动动作编号	时间范围	向心加速度平均值
一	第 100~160 秒	8.7276
三	第 300~320 秒	14.1852
四	第 350~749 秒	11.9157

由统计结果可知，第三次机动动作的向心加速度大小明显大于第一、四次机动动作。这是由于第一次机动动作是一个大范围的转弯，第四次机动动作是盘旋，而第三次机动动作是一个短时间的急转弯，这就导致向心加速度的显著增大。

5.4 小结

在本问题研究中，我们利用二阶常速 CV 模型对目标进行运动学建模，并利用卡尔曼滤波在线进行模型状态的估计。利用“Data1.txt”实际目标飞行数据进行仿真，利用三个雷达的测量数据估计出目标的轨迹、速度的变化以及加速度的大小和方向变化，并在此基础上分析了目标机动行为和发生的时间范围，目标航迹和加速度方向均以图形方式给出，目标机动动作主要如下：

(1) 第一次机动动作（完成一个约逆时针 180 度的转弯）的时间范围大致在相对时间第 100 秒到第 160 秒（绝对时间第 3.6720400e+004 秒到第 3.6780400e+004 秒）左右。

(2) 第二、三次机动动作（完成两个逆时针 90 度的转弯）的时间范围大致在相对时间第 300 秒到第 320 秒（绝对时间第 3.6920400e+004 秒到第

3.6940400e+004 秒) 左右。

(3) 第四次机动动作(逆时针盘旋两周)的时间范围大致在相对时间第 350 秒到第 749 秒(绝对时间第 3.7250400e+004 秒到第 37369400e+004 秒)左右。

特别注意的是,本部分提出的目标航迹估计和机动检测方法是一种**在线实时目标跟踪**方法,具有实际的工程实践意义。

6 问题 2：多目标航迹跟踪与数据关联

6.1 问题分析与假设

6.1.1 问题描述

问题 2 表述如下：附件中的 Data2.txt 数据对应两个目标的实际检飞考核的飞行包线（检飞：军队根据国家军标规则设定特定的飞行路线用于考核雷达的各项性能指标，因此包线是有实战意义的）。请完成各目标的数据关联，形成相应的航迹，并阐明你们所采用或制定的准则（鼓励创新）。如果用序贯实时的方法实现更具有意义。若出现雷达一段时间只有一个回波点迹的状况，怎样使得航迹不丢失？请给出处理结果。

“Data2.txt”为某个雷达站获得的两个机动目标的测量数据，测量误差如下表所示。

表 6-1 Data2 数据测量误差表

目标	测距误差(m)	方位角误差($^{\circ}$)	俯仰角误差($^{\circ}$)
1	100	0.3	0.3
2	100	0.6	0.6

本题要求根据测量数据完成两个机动目标的数据关联[13]-[20]，形成相应的航迹，并阐明所采用的准则。数据关联的问题的根本任务是针对目标特征信息的不确定性，划分传感器接收到的信息，建立起目标和量测之间的对应关系。

在此基础上，本题研究的**主要任务**包括：

（1）制定两个目标的航迹数据关联准则，对目标的实际验飞考核的飞行包线数据进行关联。

（2）设计实时在线航迹跟踪策略，跟踪两个目标的航迹。

（3）当出现雷达一段时间只有一个回波点迹的状况，设计一种方法使得航迹不丢失并给出处理结果。

6.1.2 数据初步分析

利用 MATLAB 对 Data2.txt 进行数据提取，可知数据为一个 2649×5 的数据矩阵。其中，前三列分别为两个目标的极坐标数据，第四列为两个目标的数据测量的绝对时间，第五列为传感器编号。通过数据可知，**雷达测量周期为 0.6 秒**。

经过对数据分析，发现 Data2.txt 数据存在三个问题：

- 1、同一传感器的对两个目标的测量数据混合在了一起；
- 2、传感器传回的数据中含有同一时间传回两个目标的量测信息的情况；
- 3、传感器传回的数据中在某些时间点缺失量测信息。

6.1.3 问题难点

数据关联作为确定传感器所接收量测与目标源之间关系的过程，是目标跟踪系统中最核心也是最重要的内容。正确的进行数据关联是所有目标跟踪方法的前提。如果关联错误，会直接导致对目标状态估计偏差很大，影响后续处理的精度。在单个目标单个量测的情况下，不需要进行数据关联，就能自然建立起目标与量测的关系。当有多个目标且间隔距离较近时，数据关联就显得异常重要，而且关联过程也会变得十分复杂。

通过对问题的初步分析，发现该问题的研究存在如下难点：

(1) 将传感器接收到的量测进行预选，利用先验信息和统计学知识，筛掉目标跟踪门以外的那些能够确切判定不属于目标的量测。如何设置跟踪门的大小直接对关联产生重大影响。

(2) 怎么解决当雷达出现一段时间只有一个回波点迹的情况下，使得航迹不丢失的问题。

6.1.4 问题假设

与问题 1 假设类似，本问题给出以下假设：

假设 1：在本题中，将目标各个方向的速度变化看作是具有随机扰动特性的扰动输入，并假设其服从零均值白色高斯分布。

假设 2：表 6-1 给出的雷达测量误差为测量值与真实值的最大偏差。

6.2 问题解决方案

6.2.1 解决思路

针对问题分析中本题目的三个主要任务，解决方案如下：

首先，根据假设 1 和假设 2，建立目标的运动学方程模型，其中状态变量为机动目标在直角坐标系下的位置和速度。基于假设 1，我们可利用 CV 模型来跟踪目标航迹。

其次，设计基于卡尔曼状态估计器的目标航迹跟踪器。

第三，对目标一进行建档，利用卡尔曼滤波方法估计模型状态变量，获取目标一航迹变化和目标速度变化。运用文献[5]中的最近邻(nearest neighbor, NN)方法以目标预测位置为中心，按照一定的门限值设定一个跟踪门，将位于跟踪门内的量测，作为目标状态估计的候选量测。如果出现数据不属于目标一跟踪门的情况，则为目标二建档，利用卡尔曼滤波方法估计模型状态变量，获取目标二航迹变化和目标速度变化。由于数据为极坐标形式，在进行卡尔曼状态估计过程中，需要将雷达测量数据转换到以雷达位置为原点的直角坐标系内。

最后，根据目标一和目标二的状态估计结果，形成两个目标的相应的航迹。对于雷达一段时间只有一个回波点迹的状况，我们采用限定目标航迹跟踪器状态更新周期以及放宽目标二跟踪门限的策略来应对。当有新数据来时，先让目标一进行关联，如果数据关联到目标一，则目标二利用给定的状态更新周期更新状态；如果数据没有关联到目标一，则我们更偏向认为此数据属于目标二，用于目标二的航迹跟踪。

研究思路如下图所示：

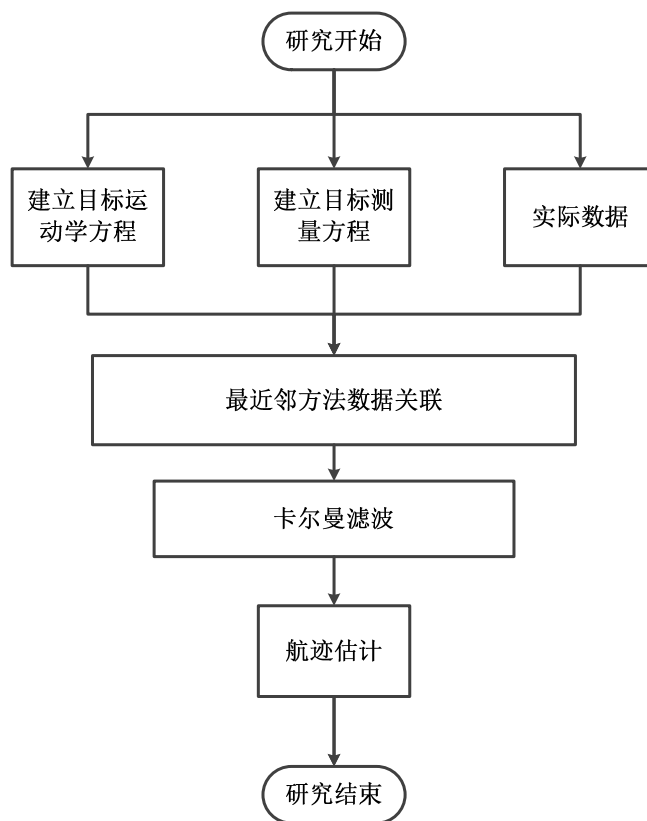


图 6-1 问题 2 研究思路

6.2.2 基于最近邻近方法的概率数据关联算法

数据关联的一般过程如图 6-2 所示：

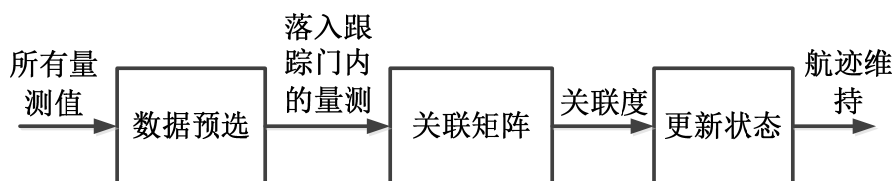


图 6-2 数据关联过程

1971 年 Singer 和 Sea 在他们以及前人工作基础上，提出了一种利用先验统计特性估计相关性的跟踪算法，即最近邻(nearest neighbor, NN)方法。该方法是一种能够在多回波环境下工作，且具有固定记忆的跟踪方法。所谓的“最近”，是表示在统计距离最小或残差概率密度最大。NN 法是以目标预测位置为中心，按照一定的门限值设定一个跟踪门，将位于跟踪门内的量测，作为目标状态估计的候选量测，限制参与相关判别的量测数，进而选择在统计意义上与目标预测位置最近的有效量测，作为目标关联对象。跟踪门是跟踪空间内的一块子区间，其大小由接收正确回波的概率来确定。判断量测是否是候选量测，也就是采用门限判断量测值 $Z(k+1)$ 是否满足：

$$d^2(z) = [Z(k+1) - \hat{Z}(k+1|k)]^T S^{-1}(k+1) [Z(k+1) - \hat{Z}(k+1|k)] \leq \gamma \quad (6.1)$$

式中， $d^2(z)$ 表示量测的残差向量范数，即量测与目标预测位置之间的统计

距离； $Z(k+1)$ 表示 $k+1$ 时刻量测值； $\hat{Z}(k+1|k)$ 表示该时刻目标的预测位置；

$S(k+1)$ 表示量测的新息协方差矩阵。

若落入跟踪门内的量测值有且仅有 1 个，则直接利用该量测值进行目标航迹更新；若落入跟踪门内的量测值有 1 个以上，则取与目标预测位置最近的量测作为与目标关联的回波信号，即使得新息向量的加权范数 $d^2(z)$ 达到极小的量测，用于对目标状态进行更新。

最近邻算法具有计算简单且计算量小，便于实现等优点，在信噪比高，目标密度小的环境下，能够得到很好的应用。在本题中，“Data2.txt”给出了两个目标的飞行包线。因此，**本题只需完成两个目标的数据关联，且杂波较少，最近邻算法可以解决本问题。**

6.2.3 构建目标跟踪模型

根据本论文第二部分，我们得到目标跟踪 CV 模型如下：

$$X(k+1) = \Phi X(k) + \Gamma w(k) \quad (6.2)$$

其中 $X = [x, y, z, v_x, v_y, v_z]^T$ ，状态转移矩阵 Φ 和系统噪声驱动矩阵 Γ 分别为：

$$\Phi = \begin{bmatrix} I_3 & TI_3 \\ 0 & I_3 \end{bmatrix}, \Gamma = \begin{bmatrix} \frac{1}{2}T^2I_3 \\ TI_3 \end{bmatrix} \quad (6.3)$$

其中 $T=0.6$ ， $w(k)$ 满足如下正态分布：

$$w \sim N(0, Q) \quad (6.4)$$

$$Q = E[\Gamma w w^T \Gamma^T] = \Gamma \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \Gamma^T \quad (6.5)$$

其中 $\sigma = [\sigma_1, \sigma_2, \sigma_3]^T$ ，

$$\sigma_1 = 0.5a_{x_max}, \sigma_2 = 0.5a_{y_max}, \sigma_3 = 0.5a_{z_max} \quad (6.6)$$

其中， $a_{x_max}, a_{y_max}, a_{z_max}$ 为各方向的最大加速度值，具体数值可在目标跟踪时根据实际效果调节，从而适应速度变化程度不同的目标
直角坐标系内的测量方程为

$$Z(k) = HX(k) + v(k) \quad (6.7)$$

其中 $H = [I_3 \ 0_{3 \times 3}]$ ，系统噪声 $v(k)$ 满足如下正态分布：

$$v(k) \sim N(0, R_L(k)) \quad (6.8)$$

其中 $R_L(k)$ 为直角坐标系下的测量噪声协方差阵，满足[3]:

$$R_L(k) = J(z_s(k)) R J^T(z_s(k)) \quad (6.9)$$

其中 Jacobian 矩阵 $J(z_s(k))$ 由实时的极坐标下测量值 $z_s(k)$ 确定:

$$J(z_s(k)) = \left. \frac{\partial h(z_s)}{\partial z_s} \right|_{z_s=z_s(k)} = \begin{bmatrix} \cos \theta \sin \varphi & r \cos \theta \cos \varphi & -r \sin \theta \sin \varphi \\ \cos \theta \cos \varphi & -r \cos \theta \sin \varphi & -r \sin \theta \cos \varphi \\ \sin \theta & 0 & r \cos \theta \end{bmatrix} \quad (6.10)$$

基于假设 2，雷达在极坐标系下对目标一的测量误差协方差阵为

$$R = R_1 = \begin{bmatrix} (100/3)^2 & 0 & 0 \\ 0 & (0.3/3)^2 & 0 \\ 0 & 0 & (0.3/3)^2 \end{bmatrix} \quad (6.11)$$

同理，对目标二的测量误差协方差阵为:

$$R = R_2 = \begin{bmatrix} (100/3)^2 & 0 & 0 \\ 0 & (0.6/3)^2 & 0 \\ 0 & 0 & (0.6/3)^2 \end{bmatrix} \quad (6.12)$$

6.2.4 目标数据关联与航迹跟踪

利用 4.2 节给出的卡尔曼状态估计器，设计目标航迹跟踪算法，并对目标航迹进行跟踪。

由于 Data2.txt 中给出的数据为两个目标的雷达测量数据，因此我们需要设计相应的航迹关联算法来进行数据关联，从而将两个目标的航迹分别辨识并跟踪出来。

考虑到雷达数据存在丢失、多目标数据混合等问题，我们不能直接利用第 4.2 节给出的卡尔曼滤波状态估计器进行目标跟踪，我们给出了修正的卡尔曼状态估计过程来跟踪目标航迹，算法步骤如图 6-3 所示:

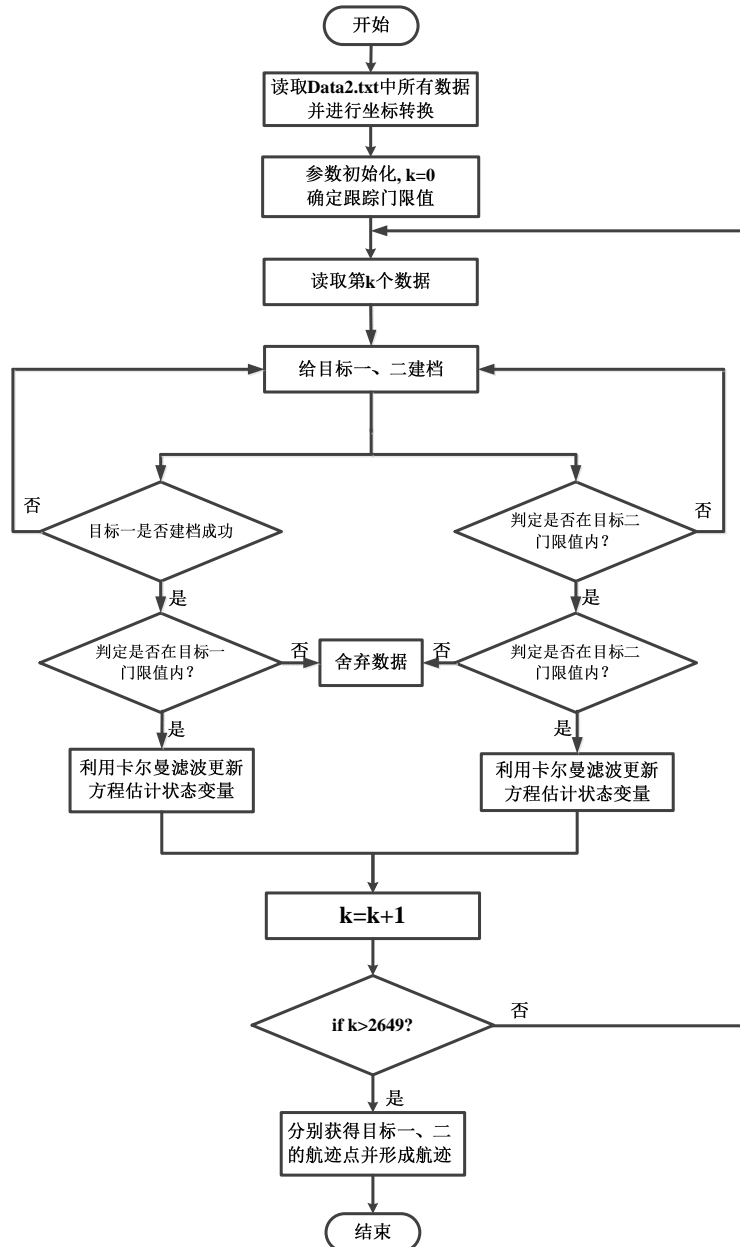


图 6-3 最近邻算法流程图

(1) 首先以目标预测位置为中心，按照一定的门限值设定一个跟踪门，其中目标二的门限值要比目标一的门限值大。

(2) 将以极坐标形式给出的量测数据转换成直角坐标系下的量测数据。

(3) 将转换后的数据分别于目标一和二的门限值比较，若既不在目标一又不在目标二的门限内，则认为该数据无效并舍弃；若在目标一的门限内则属于目标一，同理属于目标二；当同时属于目标一和二时，则属于距离两个目标的中心距离更近的那个目标。

(4) 将得到的各目标信息用于卡尔曼滤波的更新。

(5) $k = k + 1$ 。

(6) 若 $k > 2649$ ，结束滤波；否则，重复 (3) - (5)。

(1) 读取 Data2.txt 中的所有测量数据，并将极坐标形式的目标测量值转

换到以雷达位置为原点的直角坐标系，转换公式如下

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r \cos \theta \sin \varphi \\ r \cos \theta \cos \varphi \\ r \sin \theta \end{bmatrix} \quad (6.13)$$

(2) $k=0$

(3) 对于时刻 k ，读取 Data2.txt 中的第 k 个数据，并判断是否落入目标一的跟踪门内。若数据落入目标一的跟踪门内，则跳转到步骤 (4)；否则，判断数据是否落入目标二的跟踪门内，若落入目标二的跟踪门内，则跳转到步骤 (5)。

(4) 利用卡尔曼状态估计器完成目标一的状态变量的估计。

(5) 利用卡尔曼状态估计器完成目标二的状态变量的估计。

(6) 如果目标一或目标二没有关联到该数据上，并且当前数据时间与目标一或目标二的上次状态更新时间偏差大于 0.5 秒，则为了保证航迹估计的连续性，我们仅利用上一时刻的估计值对当前状态进行更新，更新公式如下

$$\begin{aligned} \hat{X}(k+1|k+1) &= \Phi \hat{X}(k|k) \\ P(k+1|k) &= \Phi P(k|k) \Phi^T + Q \\ K(k+1) &= P(k+1|k) H^T [H P(k+1|k) H^T + R_L]^{-1} \\ P(k+1|k+1) &= [I - K(k+1) H] P(k+1|k) \end{aligned} \quad (6.14)$$

(7) $k=k+1$ 。

(8) 若 $k > 2649$ ，结束滤波；否则，重复 (3) - (7)。

6.3 仿真结果与分析

运行 “\B10006002fj\MATLAB 仿真程序\问题 2\Main.m”，可以得出目标位置和速度的变化的估计值。在这里，我们取各个方向的加速度最大值为 $1g$ 。基于此，我们在仿真中取

$$\sigma = [0.5g, 0.5g, 0.5g]^T \quad (6.15)$$

相关仿真配置如下表所示：

表格 6-1 仿真配置表

序号	变量	数值
1	采样间隔 T	0.6
2	目标一的门限值 γ_1	50
3	目标二的门限值 γ_2	500

图 6-4 给出了目标的三维航迹图，其中红色星点为原始数据，蓝色圆点为航迹估计数据。为了方便读者观察其航迹，图 6-5 给出了目标航迹在 X-Y 平面上的投影。

图 6-4 给出了两个目标的三维航迹图，蓝色圆点是目标一的航迹点，红色圆圈点为目标二的航迹点，绿色星点为所有数据点。为了方便读者观察其航迹，

图 6-5 给出了两个目标航迹在 X-Y 平面上的投影; 图 6-6 给出了两个目标航迹在 X-Z 平面上的投影; 图 6-7 给出了两个目标航迹在 Y-Z 平面上的投影。

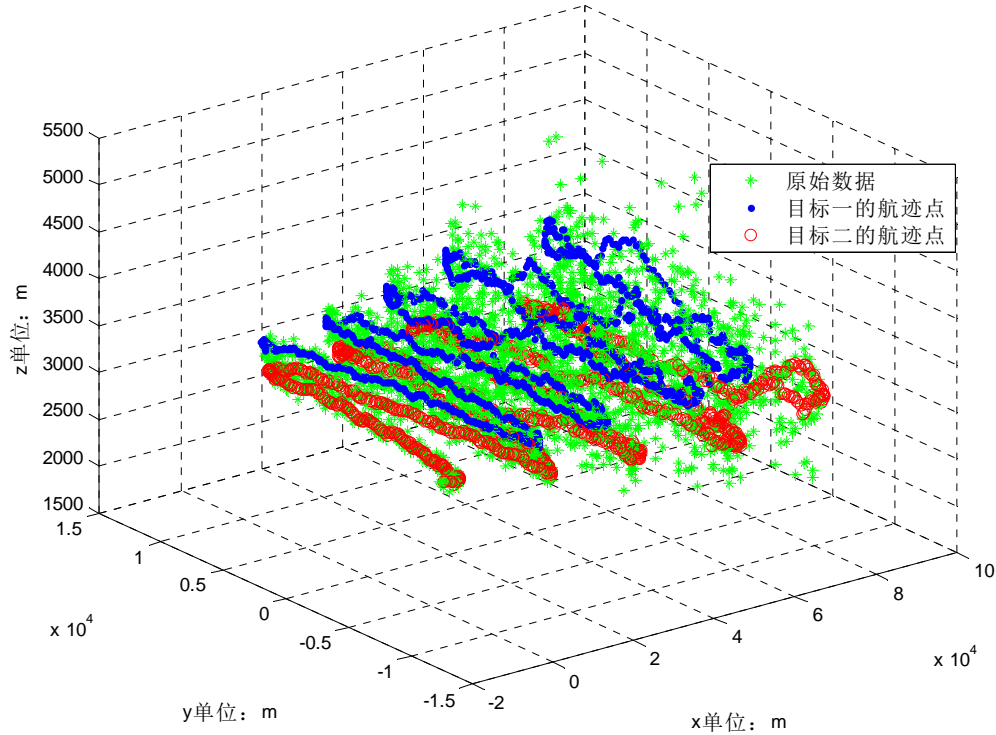


图 6-4 两个目标的三维航迹图

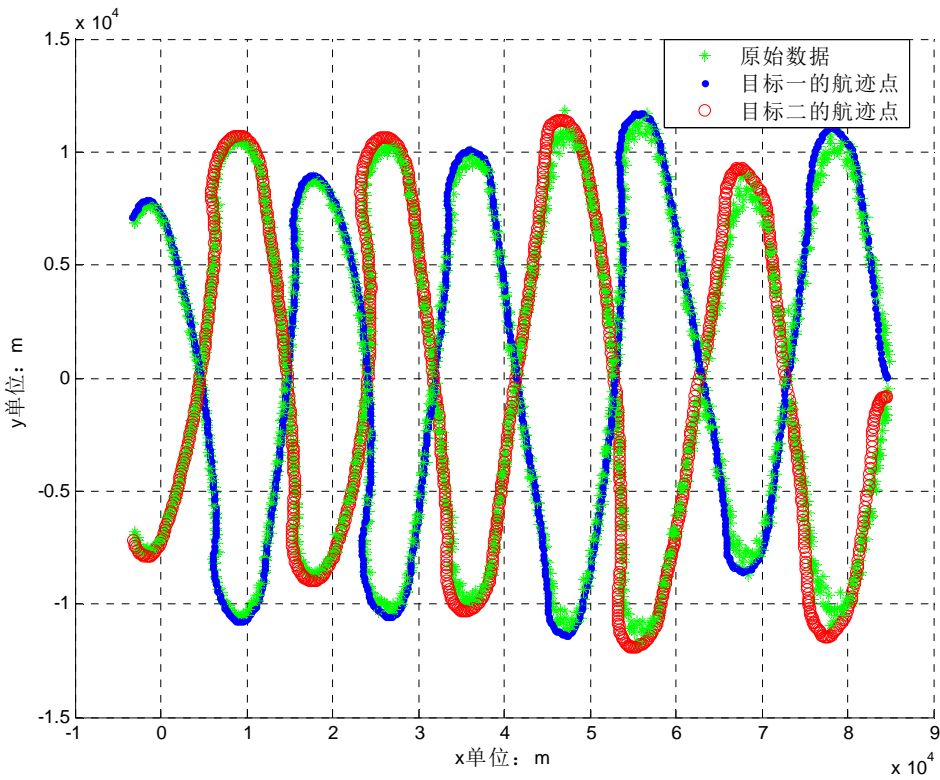


图 6-5 X-Y 平面的航迹视角图

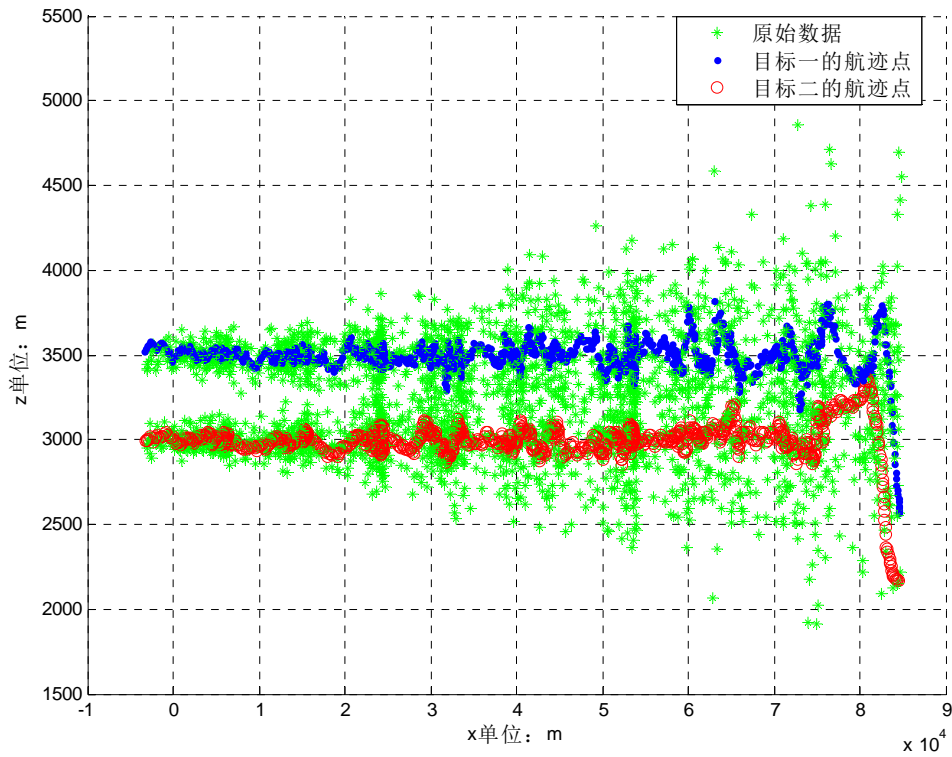


图 6-6 X-Z 平面的航迹视角图

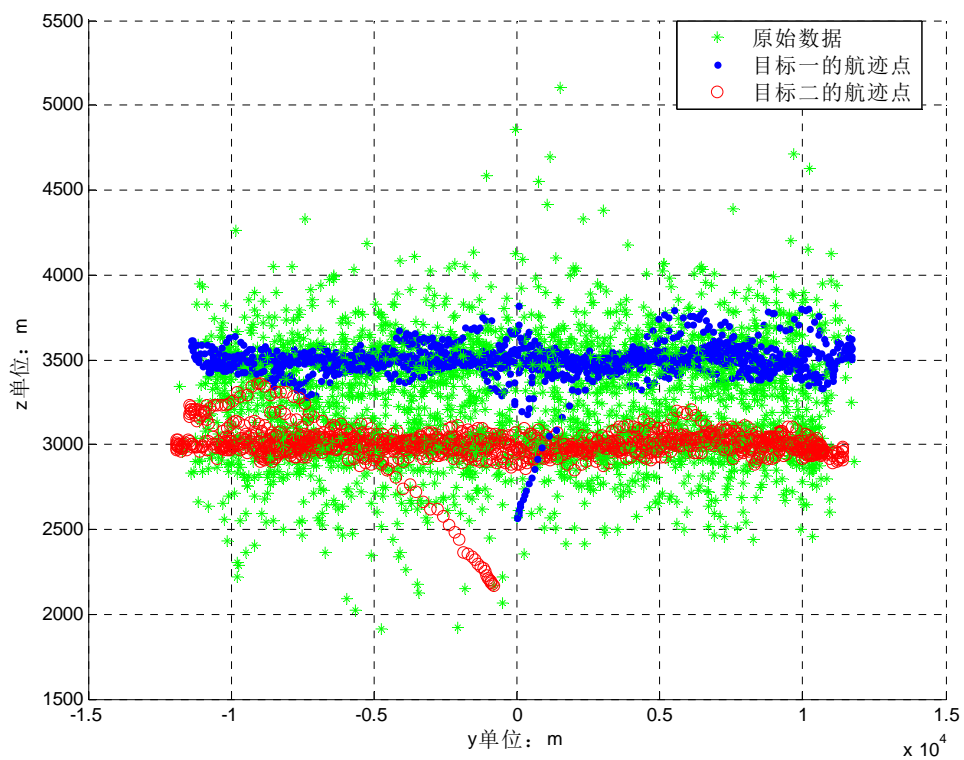


图 6-7 Y-Z 平面的航迹视角图

通过上图，根据“Data2.txt”的测量数据，并采用上述提出的最近邻数据关联的方法可以很好地将正确的信息关联到各自的目标上，并形成了各自的相

应航迹。同时可以看出是两个飞行器分别从某处飞往两个不同高度，并且在各自的高度平面上做蛇形机动。

考虑无杂波的情况下,对于若出现雷达一段时间只有一个回波点迹的状况,首先假定该单个回波点迹属于目标一,即优先考虑目标一,通过以上算法判断改点迹是否属于目标一;如若不属于目标一,则我们更偏向认为该回波点迹属于目标二。

6.4 小结

在本问题研究中,我们首先利用将“Data2.txt”实际目标飞行数据转换成直角坐标系下的飞行数据,接着利用最近邻的方法确定跟踪门的门限值,通过门限值对数据的合理处理得到各目标的候选量测值,利用二阶常速 CV 模型对目标进行运动学建模,并利用卡尔曼滤波在线进行模型状态的估计,并在此基础上分析得出了各目标的航迹。

特别注意的是,本部分提出的数据关联方法是一种**序贯实时**的方法,具有实际的工程实践意义。

7 问题 3：加速度随时间变化目标的机动变化规律分析

7.1 问题分析与假设

7.1.1 问题描述

问题 3 表述如下：根据附件中 Data3.txt 的数据，分析空间目标的机动变化规律(目标加速度随时间变化)。若采用第 1 问的跟踪模型进行处理，结果会有哪些变化。

Data3 中雷达坐标和测量误差如下表所示。

表 7-1 雷达坐标和测量误差表

经度 (°)	纬度 (°)	高度 (m)	测距误 差(m)	方位角误差 (°)	俯仰角误差 (°)
118	39.5	0	100	0.5	0.5

本题要求根据雷达测量数据完成单目标的航迹跟踪，形成相应的航迹，并分析目标的机动变化规律，并给出利用第 1 问的跟踪模型进行处理的结果分析。因此，本题研究的**主要任务**包括：

- (1) 构建新的目标跟踪模型，跟踪目标航迹。
- (2) 根据目标加速度变化情况，分析目标的机动变化规律。
- (3) 利用问题 1 中的跟踪模型进行目标航迹跟踪，分析其结果，指出模型存在的不足。

7.1.2 数据初步分析

利用 MATLAB 对 Data3.txt 进行数据提取，可知数据为一个 529×5 的数据矩阵。其中，前三列分别为目标的极坐标数据，第四列为数据测量的绝对时间，第五列为传感器编号。对第四列进行差分，可得到目标从第 0 秒到第 527 秒的测量数据，采样周期为 1 秒。通过分析“Data3.txt”的测量数据，发现该组数据具有以下特点：

- (1) 所有测量数据均为单一雷达测量，
- (2) 除最后一组数据外，所有数据为等间隔采样，采样间隔为 1 秒。

7.1.3 问题难点

根据问题描述，可知目标加速度随时间变化，所以问题 1 中所采用的 CV 模型已经不再适用，我们需要为目标建立新的航迹跟踪模型。

7.1.4 问题假设

假设 1：在本题中，将目标各个方向的**加速度**变化看作是具有随机扰动特性的扰动输入，并假设其服从零均值白色高斯分布。

基于假设 1，本问题中我们采用 CA 模型来进行目标的航迹跟踪。

假设 2：表 7-1 雷达测量误差为测量值与真实值的最大偏差。

7.2 问题解决方案

7.2.1 解决思路

针对问题分析中本题目的三个主要任务，解决方案如下：

首先，根据假设 1 和假设 2，建立目标的三阶 CA 模型，其中状态变量为机动目标在直角坐标系下的位置、速度和加速度。

其次，利用卡尔曼滤波方法估计目标状态，获取目标航迹变化、目标速度变化以及目标加速度变化。

第三，根据目标三轴加速度变化，获取目标加速度大小和方向。并综合考虑目标状态，分析目标的机动变化规律。

最后，利用问题 1 中的 CV 跟踪模型跟踪目标航迹，分析跟踪结果以及存在的问题。

研究思路如下图所示：

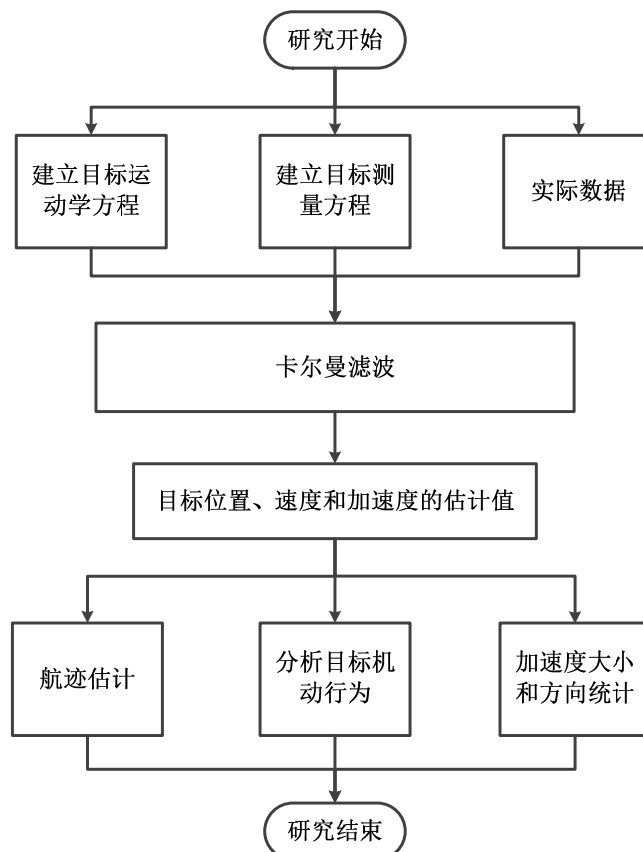


图 7-1 问题 3 研究思路

7.2.2 构建目标跟踪模型

根据本论文第二部分，我们得到目标跟踪 CA 模型如下：

$$X(k+1) = \Phi X(k) + \Gamma w(k) \quad (7.1)$$

其中 $X = [x, y, z, v_x, v_y, v_z, a_x, a_y, a_z]^T$ ，状态转移矩阵 Φ 和系统噪声驱动矩阵 Γ 分别为：

$$\Phi = \begin{bmatrix} I_3 & TI_3 & 1/2T^2I_3 \\ 0 & I_3 & TI_3 \\ 0 & 0 & I_3 \end{bmatrix}, \Gamma = \begin{bmatrix} 1/2T^2I_3 \\ TI_3 \\ I_3 \end{bmatrix} \quad (7.2)$$

其中 $T=1$ ， $w(k)$ 满足如下正态分布：

$$w \sim N(0, Q) \quad (7.3)$$

$$Q = E[\Gamma w w^T \Gamma^T] = \Gamma \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \Gamma^T \quad (7.4)$$

其中 $\sigma = [\sigma_1, \sigma_2, \sigma_3]^T$ ，

$$\sigma_1 = 0.5\Delta a_x, \sigma_2 = 0.5\Delta a_y, \sigma_3 = 0.5\Delta a_z \quad (7.5)$$

其中， $\Delta a_x, \Delta a_y, \Delta a_z$ 为各方向的加速度增量的最大值，具体数值可在目标跟踪时根据实际效果调节，从而适应加速度变化程度不同的目标。

直角坐标系内的测量方程为

$$Z(k) = HX(k) + v(k) \quad (7.6)$$

其中 $H = [I_3 \ 0_{3 \times 3} \ 0_{3 \times 3}]$ ，系统噪声 $v(k)$ 满足如下正态分布：

$$v(k) \sim N(0, R_L(k)) \quad (7.7)$$

其中 $R_L(k)$ 为直角坐标系下的测量噪声协方差阵，满足[3]：

$$R_L(k) = J(z_s(k)) R J(z_s(k))^T \quad (7.8)$$

其中 Jacobian 矩阵 $J(z_s(k))$ 由实时的极坐标下测量值 $z_s(k)$ 确定：

$$J(z_s(k)) = \left. \frac{\partial h(z_s)}{\partial z_s} \right|_{z_s = z_s(k)} = \begin{bmatrix} \cos \theta \sin \varphi & r \cos \theta \cos \varphi & -r \sin \theta \sin \varphi \\ \cos \theta \cos \varphi & -r \cos \theta \sin \varphi & -r \sin \theta \cos \varphi \\ \sin \theta & 0 & r \cos \theta \end{bmatrix} \quad (7.9)$$

极坐标下的测量噪声协方差阵 R 由雷达的测量误差确定。基于假设 2，雷达在极坐标系下的测量误差协方差阵为

$$R = R_1 = \begin{bmatrix} (100/3)^2 & 0 & 0 \\ 0 & (0.5/3)^2 & 0 \\ 0 & 0 & (0.5/3)^2 \end{bmatrix} \quad (7.10)$$

7.2.3 目标航迹跟踪

利用 4.2 节给出的卡尔曼状态估计器，设计目标航迹跟踪算法，并对目标航迹进行跟踪，算法步骤如图 7-2 所示：

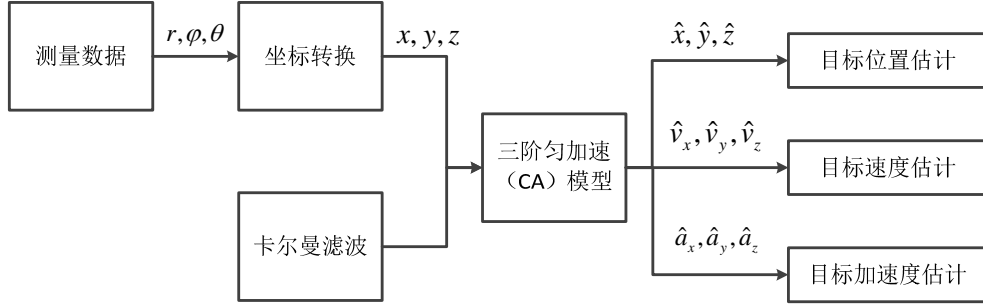


图 7-2 算法步骤

(1) 读取 Data1.txt 中的所有测量数据，并将极坐标形式的目标测量值转换到以雷达位置为原点的直角坐标系，转换公式如下

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r \cos \theta \sin \varphi \\ r \cos \theta \cos \varphi \\ r \sin \theta \end{bmatrix} \quad (7.11)$$

- (2) $k = 0$
- (3) 对于时刻 k ，直接利用卡尔曼状态估计器完成系统状态变量的估计。
- (4) $k = k + 1$ 。
- (5) 若 $k > 527$ ，结束滤波；否则，重复 (3) - (4)。

注意：本算法，仅利用当前时刻数据以及历史数据实现了目标的位置估计和速度估计，是一种在线跟踪算法。

7.2.4 机动行为检测

机动动作判定准则：加速度方向竖直向下，大小近似为重力加速度，则认为目标在竖直方向做自由落体运动。

通过目标航迹跟踪算法，可获得向量 x 的估计值。取向量 \hat{x} 中的 $[\hat{x}, \hat{y}, \hat{z}]$ 可获取目标的位置估计值；取向量 x 中的 $[\hat{v}_x, \hat{v}_y, \hat{v}_z]$ 可获取目标的速度估计值；取向量 x 中的 $[\hat{a}_x, \hat{a}_y, \hat{a}_z]$ 可获取目标的速度估计值，在此基础上分析出目标机动行为和发生的时间范围。

7.3 仿真结果与分析

运行“\B10006002fj\MATLAB 仿真程序\问题 3\Main_1.m”，可以得出目标位置和速度的变化的估计值。

在这里，我们取各个方向的加速度变化最大值为 $10g$ 。基于此，我们在仿真中取

$$\sigma = [5g, 5g, 5g]^T \quad (7.12)$$

取采样间隔为 1 秒。**注意：本题均利用相对时间绘制仿真结果图。**

图 7-3 给出了目标的三维航迹图，红色圆点为雷达测量数据，蓝色曲线为目标航迹跟踪结果。为了更直观的给出其航迹变化，图 7-4 给出了目标航迹在 X-Z 平面上的投影。

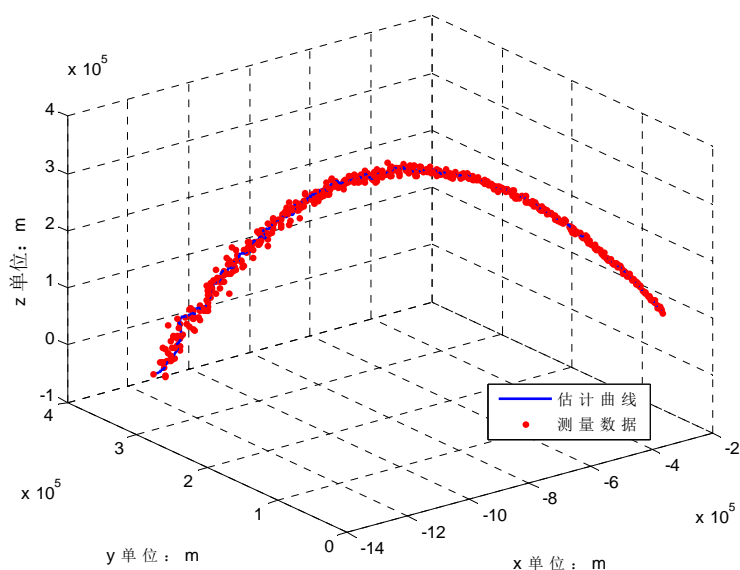


图 7-3 目标三维航迹图

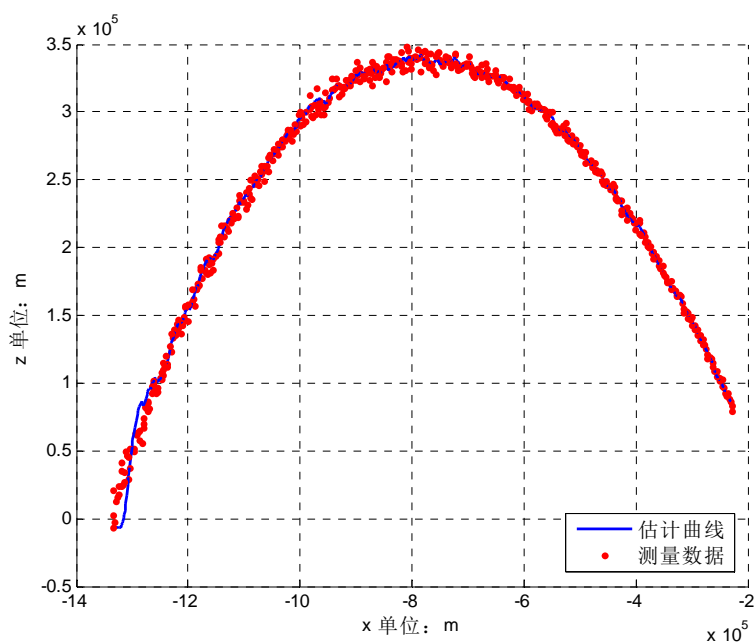


图 7-4 X-Z 平面的航迹视角图

由以上两图可以看出，该目标轨迹为一枚导弹自发射时刻起，在空中飞行的轨迹（并未着落）。目标速度和加速度变化如图 7-5 和图 7-6 所示。

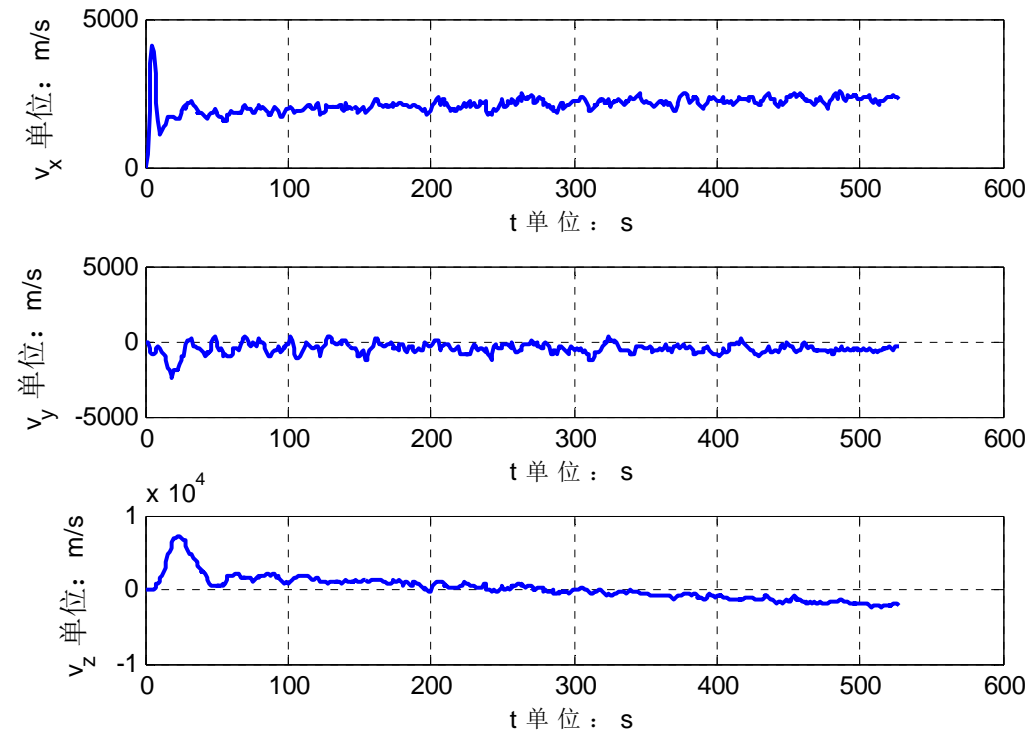


图 7-5 目标速度变化图

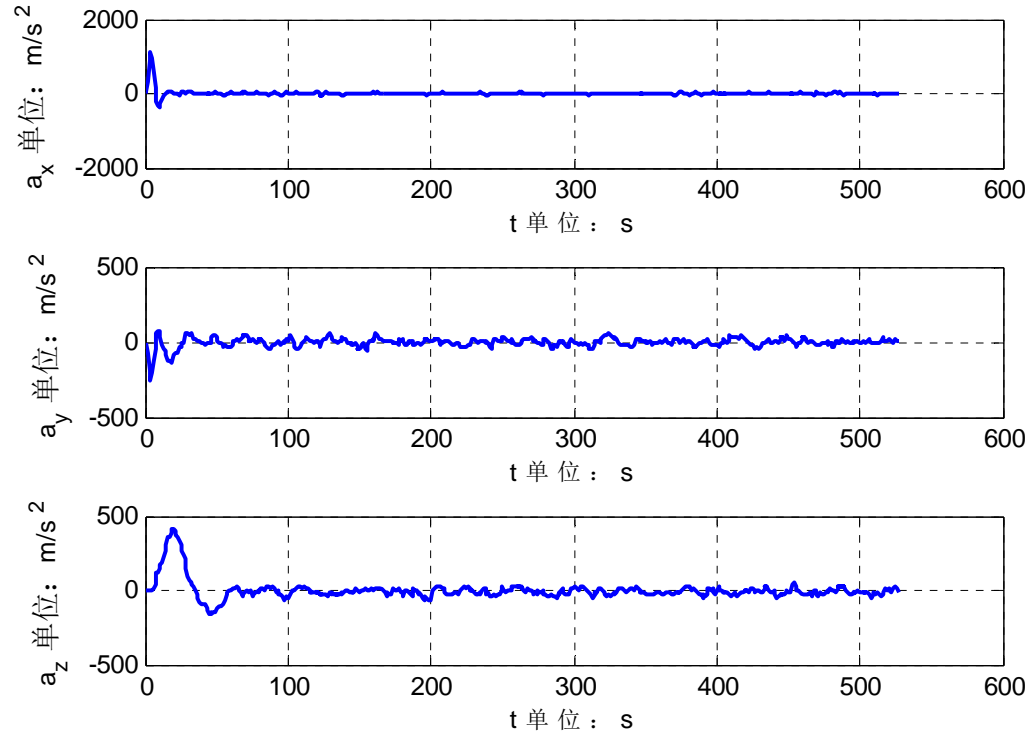


图 7-6 目标加速度变化图

对于各方向加速度的大小，我们对各方向加速度值进行统计（从第 50s 开

始统计), 结果如下表:

表格 7-1 各方向加速度大小统计结果

序号	被统计量	平均值
1	a_x	1.1195
2	a_y	0.2371
3	a_z	-10.0717

显然, 目标在 z 方向的加速度远大于在 x 和 y 方向的加速度, 接近重力加速度, 且方向指向地面。从以上结果可以看出, 目标在发射升空后, 处于近无动力飞行状态。

7.4 第 1 问中 CV 模型跟踪目标航迹结果分析

运行 “\B10006002fj\MATLAB 仿真程序\问题 3\Main_2.m”, 可以得出采用问题 1 模型来估计目标的航迹结果。图 7-7 给出了目标的三维航迹图, 图 7-8 给出了 X-Z 屏幕的目标航迹图。显然, 航迹估计结果与原始测量数据存在较大差异。这主要是因为问题 1 模型为二阶匀速 (CV) 模型, 并没有对目标加速度建模, 而本问题提出的模型为三阶匀加速 (CA) 模型, 建模中考虑了加速度的变化, 更适合估计本问题的机动变化规律 (目标加速度随时间变化)。

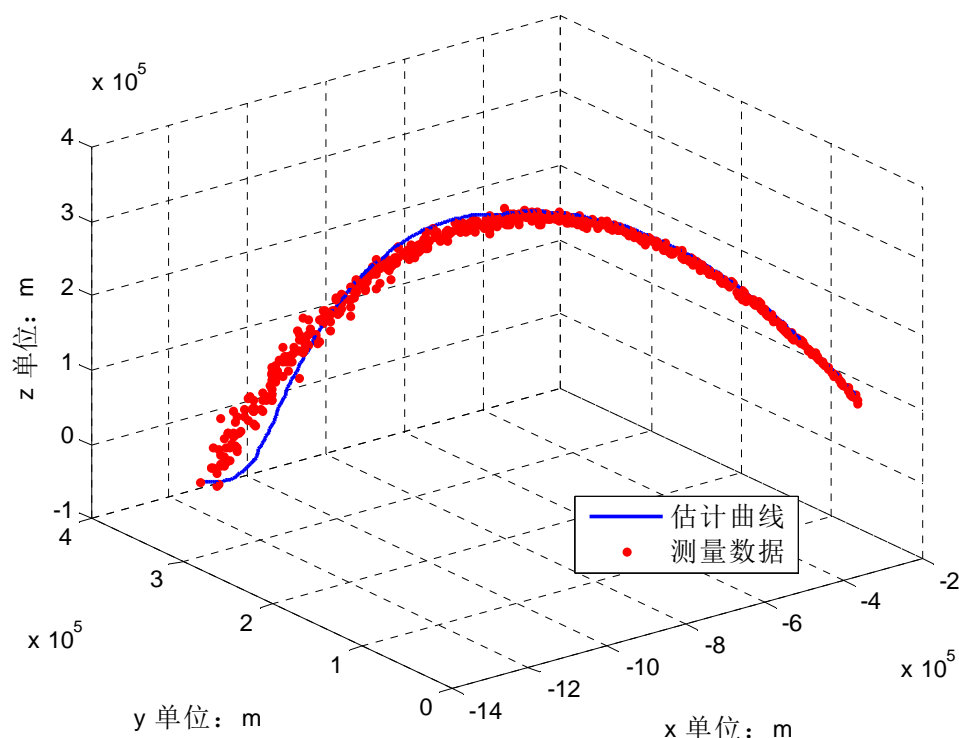


图 7-7 采用问题 1 模型估计目标三维航迹图

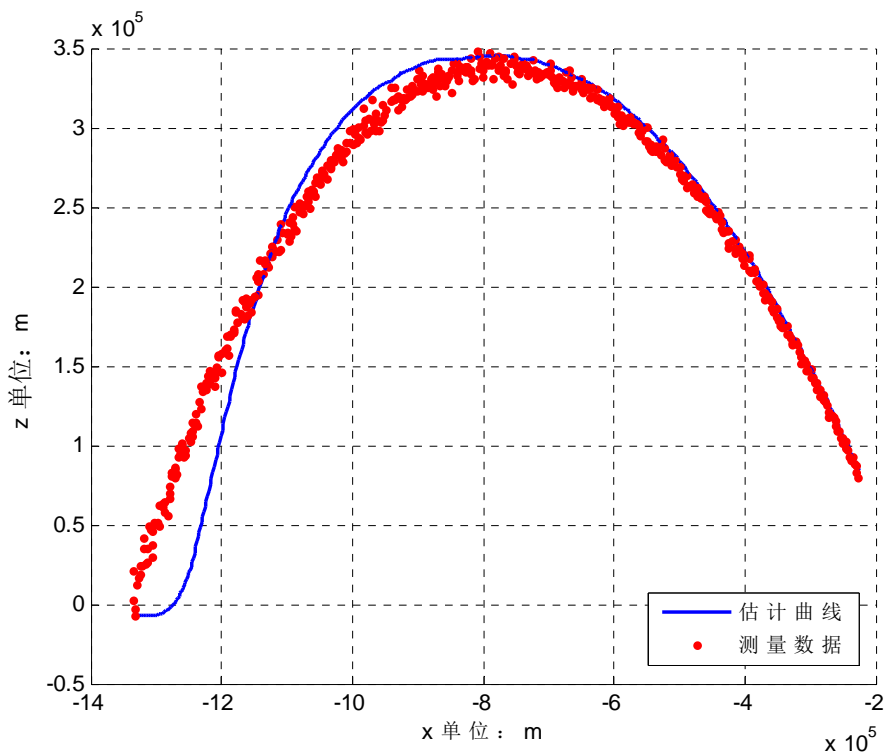


图 7-8 X-Z 平面的航迹视角图

7.5 小结

在本问题研究中，我们利用三阶匀加速（CA）模型对目标进行运动学建模，并利用卡尔曼滤波在线进行模型状态的估计。在问题仿真部分，利用“Data3.txt”实际目标飞行数据进行仿真，估计出目标的位置、速度和加速度的变化，并在此基础上分析了目标机动变化规律。根据仿真结果，该目标轨迹为一枚导弹自发射时刻起，在空中飞行的轨迹（并未着落）。

本问题的仿真部分同时也给出了基于问题 1 中的模型（二阶匀速模型）的航迹估计。从仿真结果可知，基于二阶匀速模型（CA）的航迹估计结果与原始测量数据存在较大差异，估计效果要比基于三阶匀加速(CV)模型的航迹估计效果差很多。这主要是因为问题 1 模型为二阶匀速（CV）模型，并没有对目标加速度建模，而本问题提出的模型为三阶匀加速（CA）模型，建模中考虑了加速度的变化，更适合估计本问题的机动变化规律（目标加速度随时间变化）。

特别注意的是，本部分提出的目标航迹估计和机动检测方法是一种**在线目标跟踪方法**，具有实际的工程实践意义。

8 问题 4：目标轨迹实时预测与着落点估计

8.1 问题分析与假设

8.1.1 问题描述

问题 4 表述如下：请对第 3 问的目标轨迹进行实时预测，估计该目标的着落点的坐标，给出详细结果，并分析算法复杂度。

Data3 中雷达坐标和测量误差如下表所示。

表 8-1 雷达坐标和测量误差表

经度 (°)	纬度 (°)	高度 (m)	测距误 差(m)	方位角误差 (°)	俯仰角误差 (°)
118	39.5	0	100	0.5	0.5

本问题要求根据问题 3 实时估计的历史目标轨迹，实时预测目标轨迹的变化，并利用问题 3 估计的末段目标轨迹，估计该目标的着落点的坐标，并分析算法复杂度。

因此，本题研究的**主要任务**包括：

- (1) 构建目标跟踪模型，利用卡尔曼状态预测器实时预测目标轨迹。
- (2) 根据问题 3 末段的目标轨迹估计结果，估计目标着落点坐标。
- (3) 分析算法复杂度。

8.1.2 数据初步分析

利用 MATLAB 对 Data3.txt 进行数据提取，可知数据为一个 529×5 的数据矩阵。其中，前三列分别为目标的极坐标数据，第四列为数据测量的绝对时间，第五列为传感器编号。对第四列进行差分，可得到目标从第 0 秒到第 527 秒的测量数据，采样周期为 1 秒。通过分析“Data3.txt”的测量数据，发现该组数据具有以下特点：

- (1) 所有测量数据均为单一雷达测量，
- (2) 除最后一组数据外，所有数据为等间隔采样，采样间隔为 1 秒。

8.1.3 问题难点

本问题难点如下：

- (1) 如何实现目标轨迹的精确实时预测。
- (2) 如何分析算法的复杂度。

8.1.4 问题假设

假设 1：在本题中，将目标各个方向的**加速度**变化看作是具有随机扰动特性的扰动输入，并假设其服从零均值白色高斯分布。

基于假设 1，本问题中我们采用 CA 模型来进行目标的航迹跟踪。

假设 2：表 8-1 雷达测量误差为测量值与真实值的最大偏差。

8.2 问题解决方案

8.2.1 解决思路

针对问题分析中本题目的三个主要任务，解决方案如下：

首先，参考问题 3，建立目标的三阶 CA 模型，其中状态变量为机动目标在直角坐标系下的位置、速度和加速度。

其次，利用卡尔曼状态预测器预测目标状态，预测目标轨迹，并分析该算法复杂度。

第三，利用模型状态转移方程不断迭代，最终实现着落点坐标估计，并分析算法复杂度。

8.2.2 构建目标跟踪模型

根据本论文第二部分，我们得到目标跟踪 CA 模型如下：

$$X(k+1) = \Phi X(k) + \Gamma w(k) \quad (8.1)$$

其中 $X = [x, y, z, v_x, v_y, v_z, a_x, a_y, a_z]^T$ ，状态转移矩阵 Φ 和系统噪声驱动矩阵 Γ 分别为：

$$\Phi = \begin{bmatrix} I_3 & TI_3 & 1/2T^2I_3 \\ 0 & I_3 & TI_3 \\ 0 & 0 & I_3 \end{bmatrix}, \Gamma = \begin{bmatrix} 1/2T^2I_3 \\ TI_3 \\ I_3 \end{bmatrix} \quad (8.2)$$

其中 $T=1$ ， $w(k)$ 满足如下正态分布：

$$w \sim N(0, Q) \quad (8.3)$$

$$Q = E[\Gamma w w^T \Gamma^T] = \Gamma \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \Gamma^T \quad (8.4)$$

其中 $\sigma = [\sigma_1, \sigma_2, \sigma_3]^T$ ，

$$\sigma_1 = 0.5\Delta a_x, \sigma_2 = 0.5\Delta a_y, \sigma_3 = 0.5\Delta a_z \quad (8.5)$$

其中， $\Delta a_x, \Delta a_y, \Delta a_z$ 为各方向的加速度增量的最大值，具体数值可在目标跟踪时根据实际效果调节，从而适应加速度变化程度不同的目标。

直角坐标系内的测量方程为

$$Z(k) = HX(k) + v(k) \quad (8.6)$$

其中 $H = [I_3 \quad 0_{3 \times 3} \quad 0_{3 \times 3}]$ ，系统噪声 $v(k)$ 满足如下正态分布：

$$v(k) \sim N(0, R_L(k)) \quad (8.7)$$

其中 $R_L(k)$ 为直角坐标系下的测量噪声协方差阵，满足[3]：

$$R_L(k) = J(z_s(k)) R J(z_s(k))^T \quad (8.8)$$

其中 Jacobian 矩阵 $J(z_s(k))$ 由实时的极坐标下测量值 $z_s(k)$ 确定：

$$J(z_s(k)) = \left. \frac{\partial h(z_s)}{\partial z_s} \right|_{z_s=z_s(k)} = \begin{bmatrix} \cos \theta \sin \varphi & r \cos \theta \cos \varphi & -r \sin \theta \sin \varphi \\ \cos \theta \cos \varphi & -r \cos \theta \sin \varphi & -r \sin \theta \cos \varphi \\ \sin \theta & 0 & r \cos \theta \end{bmatrix} \quad (8.9)$$

极坐标下的测量噪声协方差阵 R 由雷达的测量误差确定。基于假设 2，雷达在极坐标系下的测量误差协方差阵为

$$R = R_1 = \begin{bmatrix} (100/3)^2 & 0 & 0 \\ 0 & (0.5/3)^2 & 0 \\ 0 & 0 & (0.5/3)^2 \end{bmatrix} \quad (8.10)$$

8.2.3 目标轨迹实时预测与着落点估计

利用 4.3 节给出的卡尔曼状态预测器实时预测目标，算法步骤如图 8-1 所示：

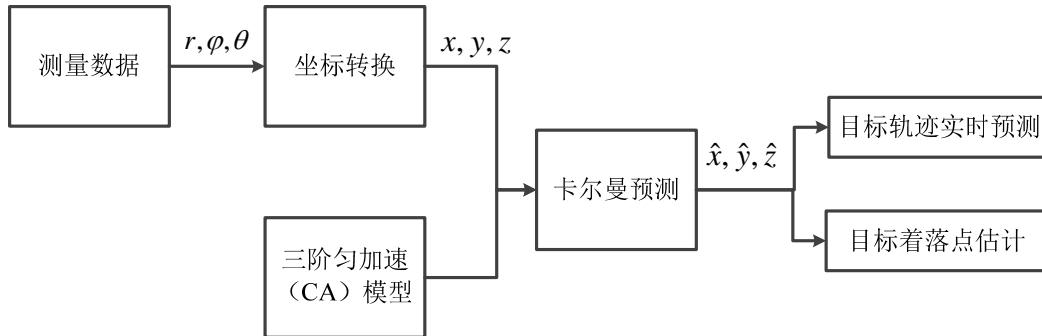


图 8-1 算法步骤

1) 读取 Data3.txt 中的所有测量数据，并将极坐标形式的目标测量值转换到以雷达位置为原点的直角坐标系，转换公式如下

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} r \cos \theta \sin \varphi \\ r \cos \theta \cos \varphi \\ r \sin \theta \end{bmatrix} \quad (8.11)$$

(2) $k=0$

(3) 对于时刻 k ，利用卡尔曼状态预测器预测目标状态。

(4) $k = k + 1$ 。

(5) 若 $k \leq 527$ ，重复 (3) - (4)。

(6) 若 $k > 527$ ，由于缺乏雷达测量数据，我们利用上一时刻的目标轨迹预测值对当前时刻目标轨迹进行预测，预测公式如下

$$\hat{X}(k+1|k) = \Phi \hat{X}(k|k-1) \quad (8.12)$$

(7) $k = k + 1$ 。

(8) 若目标高度预测值小于等于 0，则结束预测，输出目标着落点坐标；否则，重复 (6) - (7)。

注意：本算法，仅利用当前时刻数据以及历史数据实现了目标轨迹预测，是一种在线预测算法。

8.3 仿真结果与分析

运行 “\B10006002fj\MATLAB 仿真程序\问题 4\Main.m”，可以得出目标轨迹变化的预测曲线。

在这里，我们取各个方向的加速度变化最大值为 $10g$ 。基于此，我们在仿真中取

$$\sigma = [5g, 5g, 5g]^T \quad (8.13)$$

取采样间隔为 1 秒。**注意：**本题均利用相对时间绘制仿真结果图。

图 8-2 给出了基于卡尔曼预测器的目标轨迹预测结果图。

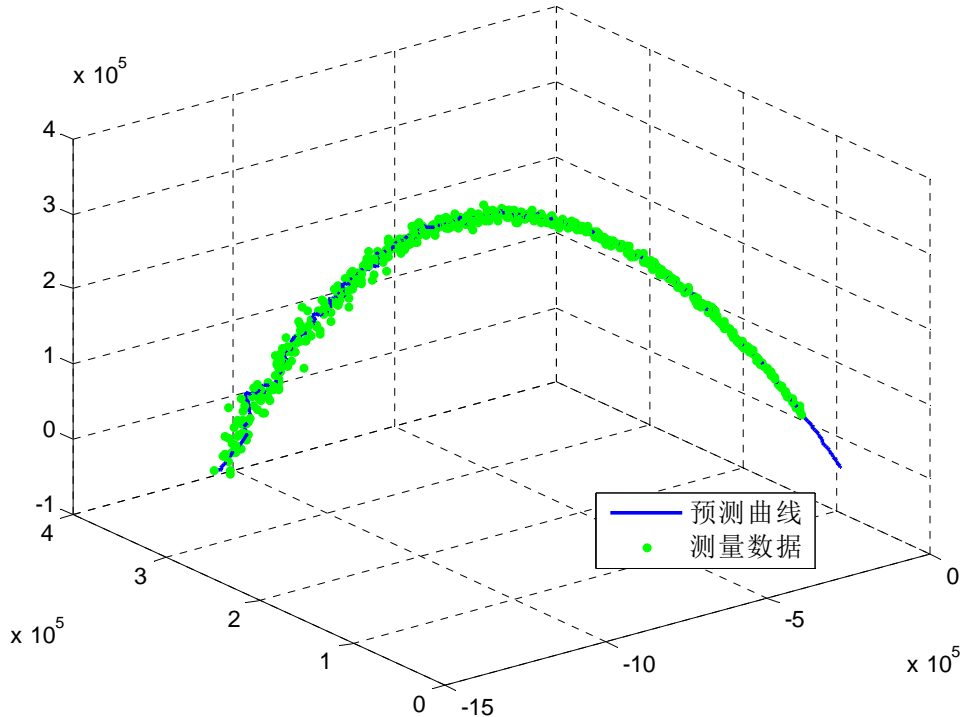


图 8-2 目标轨迹预测结果图

为了更直观的给出航迹预测结果，图 8-3 给出了目标航迹在 X-Z 平面上的投影。

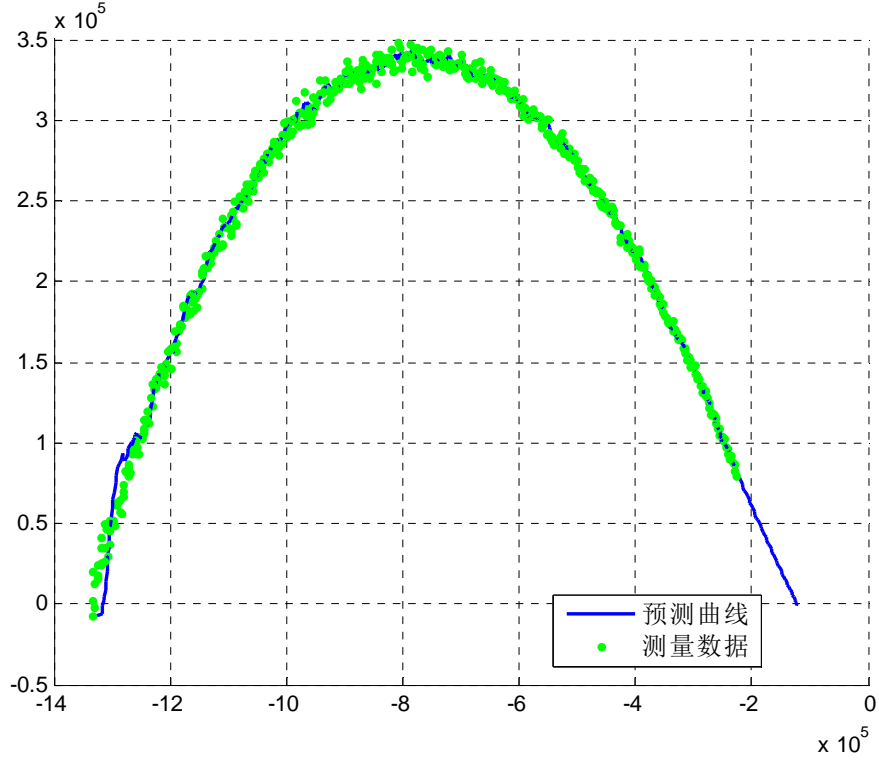


图 8-3 X-Z 平面目标轨迹预测结果图

经过对预测数据的插值，可得到着落点在以雷达为坐标原点的直角坐标系下的坐标约为

$$(x, y, z) = (-1.574029053792917e+05, 5.121891946508928e+04, 0) \quad (8.14)$$

由表 8-1 可知，雷达的经度为 118° ，纬度为 39.5° 。那么，我们有

$$(x, y, z) = \left(\frac{(\alpha - 118)\pi}{180} R_{earth} \cos\left(\frac{39.5\pi}{180}\right), \frac{(\beta - 39.5)\pi}{180} R_{earth}, 0 \right) \quad (8.15)$$

其中， (α, β) 为着落点经纬度坐标。联立式(8.14)(8.15)计算出目标着落点的经纬度坐标 $(\alpha, \beta) = (116.1738^\circ, 39.9585^\circ)$ ，大概在北京市石景山区八大处公园附近。

8.4 算法复杂度计算分析

分析算法的复杂度，最为有效的方法就是对其浮点操作数目 (floating-point-operations, flops) 进行准确统计[34]。一次 flops 定义为两个浮点数间进行一次加、减、乘或除法的运算，是算法复杂度的一种定量表述。但是应该注意到滤波中的很多运算难以用加、减、乘、除来简单表示，例如开方、数值分解、指数运算等。并且这些算法的具体实现形式呈现多样化，难易不一，因此只能将其等效为相同运行时间的 flops，所以这种统计也称为等效 flops 复杂度分析[35]。尽管是一种等效，但对表征算法的相对复杂度已足够精确了。进行分析之前首先给出滤波中常用的基本代数运算所需的 flops 次数[36]：

1) 矩阵加减法

$A \in \mathbf{R}^{n \times m}, B \in \mathbf{R}^{n \times m}$ ，计算 $A \pm B$ 需要 nm 次 flops

2) 矩阵相乘

$A \in \mathbf{R}^{n \times m}, B \in \mathbf{R}^{m \times l}$ ，计算 AB 的 flops 为 $2mnl - nl$

3) 矩阵求逆

$A \in \mathbf{R}^{n \times n}$ ， A^{-1} 的 flops 数为 n^3

4) Cholesky 分解

$A \in \mathbf{R}^{n \times n}$ ，则 $chol(A)$ 需要进行 $\frac{1}{3}n^3$ 次 flops 运算。

利用上述诸元素即可对卡尔曼状态估计器和卡尔曼状态预测器进行复杂度分析。具体的卡尔曼状态估计器和预测器的算法步骤见下表所示：

表 8-2 卡尔曼状态估计器算法

预测：

$$\hat{X}(k+1|k) = \Phi \hat{X}(k|k)$$

$$P(k+1|k) = \Phi P(k|k) \Phi^T + Q_L$$

更新：

$$\hat{Z}(k+1) = H \hat{X}(k+1|k)$$

$$P_{z_{k+1}} = H P(k+1|k) H^T + R_L$$

$$P_{x_{k+1} z_{k+1}} = P(k+1|k) H^T$$

$$K(k+1) = P_{x_{k+1} z_{k+1}} P_{z_{k+1}}^{-1}$$

$$P(k+1|k+1) = P(k+1|k) - K(k+1) P_{z_{k+1}}^{-1} K^T(k+1)$$

$$\hat{X}(k+1|k+1) = \hat{X}(k+1|k) + K(k+1) (Z(k+1) - \hat{Z}(k+1))$$

表 8-3 卡尔曼状态预测器算法

预测：

$$\hat{Z}(k) = H \hat{X}(k|k-1)$$

$$P_{z_k} = H P(k|k-1) H^T + R_L$$

$$P_{x_k z_k} = P(k|k-1) H^T$$

$$K(k) = \Phi P_{x_k z_k} P_{z_k}^{-1}$$

$$P(k+1|k) = \Phi P(k|k-1) \Phi^T - K(k) H P(k|k-1) \Phi^T + Q_L$$

$$\hat{X}(k+1|k) = \Phi \hat{X}(k|k-1) + K(k) [Z(k) - \hat{Z}(k)]$$

1) 卡尔曼计算复杂度分析

按照表 8-2 的滤波步骤, 卡尔曼状态估计器算法的复杂度可作如下的分析:

a) 状态方程预测中, 均值预测 $\hat{X}(k+1|k)$ 需要 $2n^2 - n$ 次 flops, 协方差预测

$P(k+1|k)$ 需要 $4n^3 - n^2$ 次 flops;

b) 量测方程预测中, 量测预测 $\hat{Z}(k+1)$ 需要 $2nl - l$, 量测协方差 $P_{z_{k+1}}$ 预测需要 $2n^2l - 2nl^2 - nl$ 次 flops, 量测互协方差 $P_{x_{k+1}z_{k+1}}$ 预测需要 $2n^2l - nl$ 次 flops;

c) 计算后验估计, 增益 $K(k+1)$ 需 $l^3 + 2nl^2 - nl$ 次 flops, 协方差更新

$P(k+1|k+1)$ 需要 $2nl^2 + 2n^2l - nl$ 次 flops, 状态估计 $\hat{X}(k+1|k+1)$ 需要 $2nl + l$ 次 flops。

由上述分析可得到该算法复杂度为 $4n^3 + 6n^2l + 6nl^2 + n^2 - n + l^3$, 在滤波过程中矩阵的乘运算为主要耗时因素。同理, 可以得出卡尔曼状态预测器算法复杂度为 $4n^3 + 8n^2l + 4nl^2 - n^2 - n + l^3$ 。

为了给出更加直观的结论, 运行 “\B10006002fj\MATLAB 仿真程序\问题 4\ computational_complexity_1.m” 和 “\B10006002fj\MATLAB 仿真程序\问题 4\ computational_complexity_2.m”, 分别绘出状态维数 n 和量测维数 l 变化时卡尔曼状态估计器和预测器的变化曲线如下图所示。

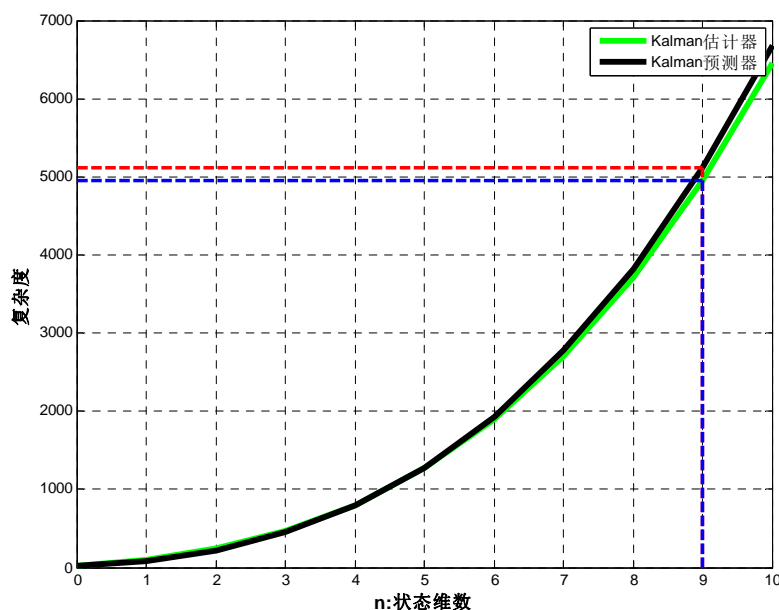


图 8-4 卡尔曼状态估计器和预测器的复杂度 ($l=3$)

局部细节图如下所示：

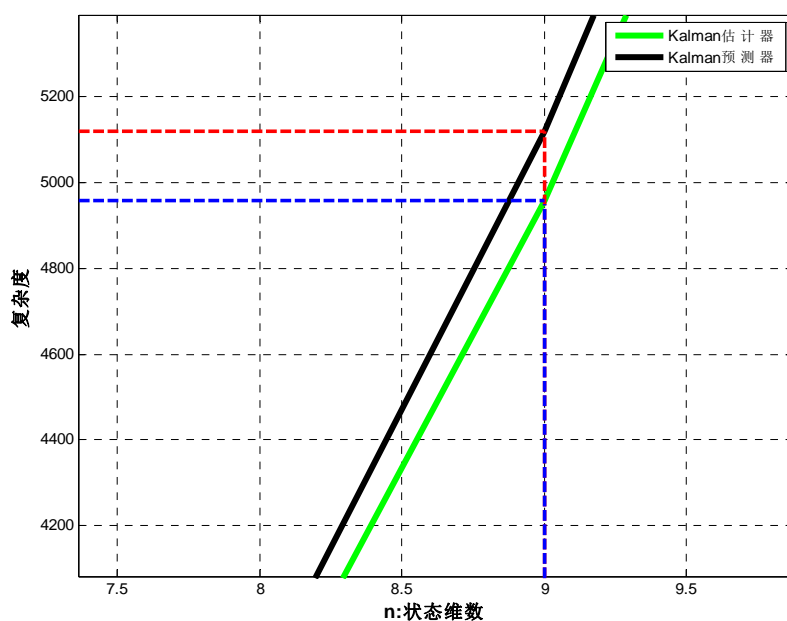


图 8-5 卡尔曼估计器和预测器的复杂度 ($l=3$) 局部细节图

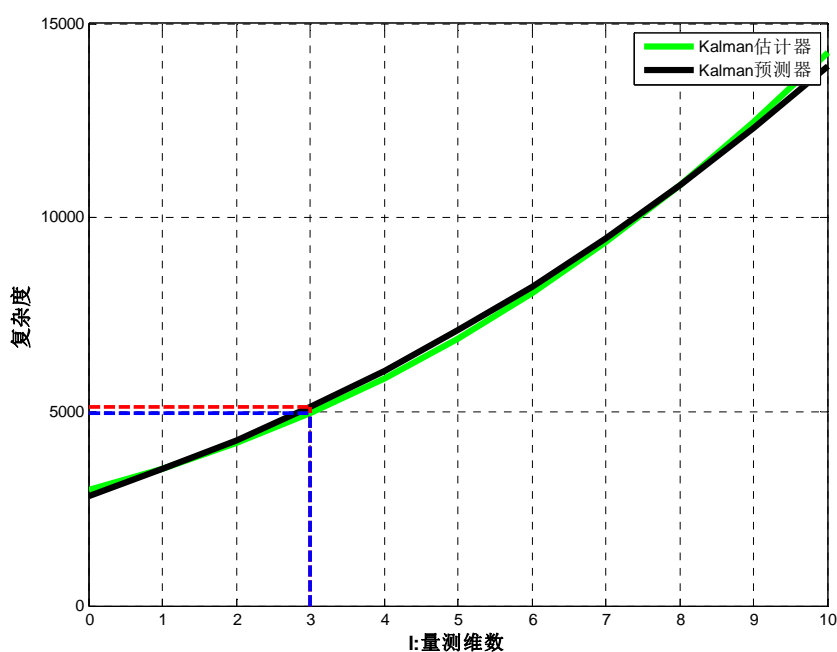


图 8-6 卡尔曼估计器和预测器的复杂度 ($n=9$)

图 8-4 是量测值维数 $l=3$ 时的卡尔曼状态估计器和预测器的复杂度，并可计算得到量测值维数 $l=3$ 时，卡尔曼状态估计器和预测器的复杂度分别是 4959 和 5121；图 8-6 是状态维数 $n=9$ 时的卡尔曼状态估计器和预测器的复杂度。

由于卡尔曼滤波器是一种递归算法，它对当前时刻值的估计值只依赖于前一个时刻。因此在程序运行的过程中，内存中只需保存前一个时刻的值即可，这充分体现了卡尔曼滤波器轻量级的特点。

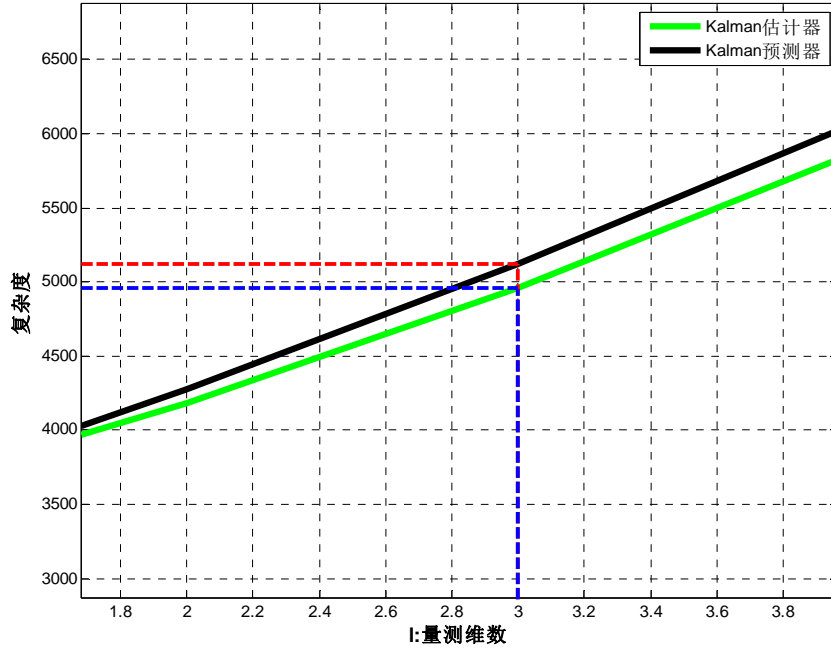


图 8-7 卡尔曼估计器和预测器的复杂度 ($n=9$)

假设当前时刻为 k ，根据改进后的卡尔曼滤波器算法，内存中需要存储的变量依次为： $\hat{X}(k+1|k+1)$ 、 $\hat{X}(k+1|k)$ 、 $P(k+1|k)$ 、 $P(k+1|k+1)$ 、 Q_L 、 $R_L(k)$ 、 $\tilde{z}(k+1)$ 、 $K(k+1)$ 等由于在程序运行的过程中变化极小，一般作为常量存储。

由此可见，卡尔曼滤波器的这种递归方式使得其在网络异常检测的应用中只需占用很小的内存空间，而不会对网络节点造成额外的负担，从而以最小的代价获得了性价比最高的异常检测模式。因此可以说，本题中的卡尔曼状态估计器和预测器具有较低的空间复杂度和时间复杂度。

8.5 小结

在本问题研究中，我们参考问题 3，建立目标的三阶 CA 模型，并利用卡尔曼状态预测器预测目标状态，预测目标轨迹。在问题仿真部分，利用“Data3.txt”实际目标飞行数据进行仿真，实现了基于卡尔曼状态预测器的目标轨迹预测和着落点估计，并分析了算法复杂度。由仿真结果可知，目标着落点的经纬度坐标为 $(116.1738^\circ, 39.9585^\circ)$ ，大概在北京市石景山区八大处公园附近。

经过等效 flops 复杂度分析，可以得到卡尔曼状态估计器和预测器的算法分别需要 4959 次和 5121 次 flops。同时，本题中的卡尔曼状态估计器和预测器具有较低的空间复杂度和时间复杂度。

特别注意的是，本部分提出的目标航迹预测方法是一种**在线目标跟踪方法**，具有实际的工程实践意义。

9 问题 5：目标的跟踪与反跟踪

9.1 问题分析

问题 5 表述如下：Data2.txt 数据中的两个目标已被雷达锁定跟踪。在目标能够及时了解是否被跟踪，并已知雷达的测量精度为雷达波束宽度为 3° ，即在以雷达为锥顶，雷达与目标连线为轴，半顶角为 1.5° 的圆锥内的目标均能被探测到；雷达前后两次扫描时间间隔最小为 0.5s。为应对你们的跟踪模型，目标应该采用怎样的有利于逃逸的策略与方案？反之为了保持对目标的跟踪，跟踪策略又应该如何相应地变换。

本问题要求根据问题 2 的解决方案、所建立的模型以及仿真分析结果，完成以下三个主要任务：

- (1) 分析总结问题 2 中跟踪模型的不足。
- (2) 根据雷达参数，确定雷达跟踪目标时的测量范围。
- (3) 针对问题 2 中跟踪模型的不足之处，以及雷达跟踪目标时的测量范围，给出目标应该采用的逃逸策略与方案。
- (4) 针对目标逃逸策略与方案，给出雷达跟踪策略调整方案。

9.2 跟踪模型特性分析

根据问题 2 的假设 1，将目标各个方向的速度变化看作是具有随机扰动特性的扰动输入，并假设其服从零均值白色高斯分布，我们采用了二阶 CV 模型来跟踪两个目标航迹，并通过限定跟踪航迹点更新周期以及放宽目标而的跟踪门限值来保证两个目标航迹不丢失。问题 2 中，我们给出的跟踪策略具有以下不足：

- (1) 二阶 CV 模型的特性，决定了不能很好的跟踪加速度随时间变化的机动目标；
- (2) 跟踪策略假定只有两个目标需要跟踪，利用最近邻数据关联方法关联目标航迹，并假设所有数据为目标数据而非杂波。

因此，如果两目标位置接近，同时做变加速度机动，则可以摆脱雷达的跟踪。

9.3 雷达锁定范围分析

跟踪雷达主要用于武器控制和为武器系统连续提供对目标的指示数据，也用于导弹靶场测量等方面。跟踪雷达首先面临的任务是捕获目标。捕获目标虽然不是跟踪工作，但在典型的雷达里，这是实现距离跟踪和角跟踪的第一步。它可以足够精确地把天线波束引向目标或者要求跟踪器扫描一个较大的可疑区域。为了快速准确地捕获目标，需要有性能良好的检测系统。

一旦目标被找到，雷达就进入跟踪状态。跟踪的参数有距离、角度和速度。距离跟踪使雷达只关注目标当前距离附近的回波，排除了对其他距离上的目标测量。对这个距离上的回波，雷达不但连续测量距离，而且连续测量目标的角度，通过自动控制系统使雷达天线随着目标运动而转动，始终指向目标。

已知雷达的测量精度为雷达波束宽度为 3° ，即在以雷达为锥顶，雷达与目标连线为轴，半顶角为 1.5° 的圆锥内的目标均能被探测到；雷达前后两次扫描时间间隔最小为 $0.5s$ 。由此可得出结论：

(1) 如果目标 1 与雷达连线和目标 2 与雷达连线之间的夹角大于 3° ，则雷达不能同时锁定两个目标；

(2) 若雷达不能同时锁定两个目标，则需要以特定区域扫描的方式分别搜寻两个目标，而雷达前后两次扫描时间间隔为 0.5 秒。

(3) 假设雷达已经锁定目标，测得目标距离为 r ，目标俯仰 θ ，目标方位 φ 。此时雷达以雷达位置为原点的直角坐标系内，能够锁定目标范围边界为一个锥形面，以目标为球心并且与此锥形面向切的内切球半径为

$$r_{\text{radar}} = r \sin\left(\frac{1.5\pi}{180}\right) \quad (9.1)$$

由此可得出结论：目标距离雷达越近，目标距离雷达锁定范围边界的距离越小，也越容易逃脱锁定。

因此，目标向垂直于与雷达连线的方向快速机动，最有可能摆脱雷达的锁定。

9.4 目标逃逸策略与方案

根据第 9.2 节与第 9.3 节的分析，目标可以采用以下策略逃逸：

(1) 进行变加速度机动动作。因为我们采用的目标跟踪模型时匀速 (CV) 运动模型，若两个目标及时了解到已被跟踪，可以通过大机动的方法来规避雷达的跟踪。

(2) 两目标以垂直于目标与雷达连线的速度，同时向相反的方向快速机动，防止雷达同时锁定两个目标。

(3) 由于我们采用的最近邻数据关联算法，因此目标采用发射干扰雷达的电磁波，使得雷达接收到的回波信号中含有大量的杂波，两个目标在三维空间中轨迹相互交叉，进行变加速机动，使得最近邻的数据关联方法失效；

(4) 目标可以发射一些干扰弹，使得单雷达接收到多目标量测信息。

综上，我们给出被雷达锁定的两个目标的逃逸策略：两目标分别释放两个干扰弹，同时以垂直于目标与雷达连线的速度方向，逆向分别进行变加速度机动，或者在三维空间中轨迹相互交叉，进行变加速机动。

9.5 跟踪策略改进

问题 2 中我们采用二阶匀速模型以及最近邻数据关联算法，从而使得目标很容易摆脱雷达锁定跟踪状态，因此需要改变目标跟踪模型。

目标运动模型时多目标跟踪技术研究的一个重要内容，目标模型建立的好坏对目标跟踪的过程影响非常大，典型的目标运动模型有：匀速和匀加速运动模型、转弯模型、圆周运动模型、时间相关模型、Markov 模型、机动目标“当前”统计模型等。对于多目标的跟踪，常用的数据关联算法有：最近邻算法、概率数据关联算法以及联合概率数据关联算法。

为了保持对目标的跟踪，可以采用交互多模型算法。多模型算法主要是用于特性随时间变化的系统的状态估计，它特别适用于机动目标。在 IMM 方法

中，假定有有限多个目标模型存在，每个模型对应于不同机动输入水平。在计算出各模型为正确的后验概率之后，就可以通过对各模型正确时的状态估计加权求和来给出最终的目标状态估计，加权因子为模型正确的后验概率。

最近邻算法和概率数据关联算法一般用于单目标或稀疏的多目标的跟踪，跟踪效果概率数据关联远优于最近邻算法；联合概率数据关联算法能够对多个已知的目标进行有效的跟踪，但是其计算量大，并且如果回波密度继续增大，其计算量将迅速增大，致使计算负荷出现组合爆炸现象，这也是联合概率数据关联的致命弱点。

可以将交互多模型与其他数据关联，如概率数据关联结合构成 IMM-PDA 算法用于单个机动目标跟踪或稀疏的多机动目标跟踪。将交互多模型与联合概率数据关联算法结合构成 IMM-JPDA 跟踪多个机动目标。

因此，为了保证对两个目标的跟踪，我们需要将跟踪模型改为交互多模型 IMM，并且将数据关联算法改为联合概率数据关联算法。

9.6 小结

在本题的研究中，本论文首先分析了问题 2 中所采用的跟踪模型的不足，并根据雷达参数，确定了雷达跟踪目标时的锁定范围。由此，针对跟踪模型的不足以及雷达的锁定范围，给出了目标应该采用的逃逸策略与方案。针对此逃逸方案，我们给出了跟踪模型的改进思路以及跟踪策略调整。

10 总结

本论文根据《2014 年全国研究生数学建模竞赛 B 题：机动目标跟踪与反跟踪》提供的资料和数据，对题目中的五个问题进行了问题分析，并分别以章节形式给出了各个问题的建模过程、解决方案以及仿真结果分析。

本论文所采用的目标跟踪模型有两种：二阶匀速模型和三阶匀加速模型。目标航迹采用卡尔曼状态估计器进行估计，采用卡尔曼状态预测器进行预测。

针对问题 1，我们利用二阶常速 CV 模型对目标进行运动学建模，并利用卡尔曼滤波在线进行模型状态的估计。利用“Data1.txt”实际目标飞行数据进行仿真，利用三个雷达的测量数据估计出目标的轨迹、速度的变化以及加速度的大小和方向变化，并在此基础上分析了目标机动行为和发生的时间范围，目标航迹和加速度方向均以图形方式给出，目标机动动作主要如下：

(1) 第一次机动动作（完成一个约逆时针 180 度的转弯）的时间范围大致在相对时间第 100 秒到第 160 秒（绝对时间第 $3.6720400e+004$ 秒到第 $3.6780400e+004$ 秒）左右。

(2) 第二、三次机动动作（完成两个逆时针 90 度的转弯）的时间范围大致在相对时间第 300 秒到第 320 秒（绝对时间第 $3.6920400e+004$ 秒到第 $3.6940400e+004$ 秒）左右。

(3) 第四次机动动作（逆时针盘旋两周）的时间范围大致在相对时间第 350 秒到第 749 秒（绝对时间第 $3.7250400e+004$ 秒到第 $3.7369400e+004$ 秒）左右。

针对问题 2，我们首先利用将“Data2.txt”实际目标飞行数据转换成直角坐标系下的飞行数据，接着利用最近邻的方法确定跟踪门的门限值，通过门限值对数据的合理处理得到各目标的候选量测值，利用二阶常速 CV 模型对目标进行运动学建模，并利用卡尔曼滤波在线进行模型状态的估计，并在此基础上分析得出了各目标的航迹。通过航迹跟踪发现，两个目标分别做交叉蛇形机动，从远距离点向雷达所在位置快速机动。

针对问题 3，我们利用三阶匀加速（CA）模型对目标进行运动学建模，并利用卡尔曼滤波在线进行模型状态的估计。在问题仿真部分，利用“Data3.txt”实际目标飞行数据进行仿真，估计出目标的位置、速度和加速度的变化，并在此基础上分析了目标机动变化规律。根据仿真结果，该目标轨迹为一枚导弹自发射时刻起，在空中飞行的轨迹（并未着落）。从仿真结果可知，基于二阶匀速模型（CA）的航迹估计结果与原始测量数据存在较大差异，估计效果要比基于三阶匀加速(CV)模型的航迹估计效果差很多。这主要是因为问题 1 模型为二阶匀速（CV）模型，并没有对目标加速度建模，而本问题提出的模型为三阶匀加速（CA）模型，建模中考虑了加速度的变化，更适合估计本问题的机动变化规律（目标加速度随时间变化）。

针对问题 4，我们参考问题 3，建立目标的三阶 CA 模型，并利用卡尔曼状态预测器预测目标状态，预测目标轨迹。在问题仿真部分，利用“Data3.txt”实际目标飞行数据进行仿真，实现了基于卡尔曼状态预测器的目标轨迹预测和着落点估计，并分析了算法复杂度。由仿真结果可知，目标着落点的经纬度坐标

为 $(116.1738^{\circ}, 39.9585^{\circ})$ ，大概在北京市石景山区八大处公园附近。

针对问题 5，本论文首先分析了问题 2 中所采用的跟踪模型的不足，并根据雷达参数，确定了雷达跟踪目标时的锁定范围。由此，针对跟踪模型的不足以及雷达的锁定范围，给出了目标应该采用的逃逸策略与方案。针对此逃逸方案，我们给出了跟踪模型的改进思路以及跟踪策略调整。

综上所述，本论文对 B 题所提出的所有问题均一一进行了解答，并相应的给出了解决方案与仿真结果分析。

参考文献

- [1] X. R. Li and V. P. Jilkov, "Survey of Maneuvering Target Tracking, Part I: Dynamic Models," IEEE Transactions on Aerospace and Electronic Systems, 39(4): 1333–1364, October 2003.
- [2] X. R. Li and V. P. Jilkov, "Survey of Maneuvering Target Tracking. Part II: Motion Models of Ballistic Targets," IEEE Transactions on Aerospace and Electronic Systems, 46(1):96–119, January 2010.
- [3] X. R. Li and V. P. Jilkov, "A Survey of Maneuvering Target Tracking—Part III: Measurement Models," In: Proceedings of SPIE - Signal and Data Processing of Small Targets 2001, 31 July–2 August 2001, San Diego, CA, USA, pp. 423–446.
- [4] 陈新海. 最佳估计理论. 北京航空航天大学出版社.
- [5] 曾斯. 多机动目标跟踪中数据关联算法的研究. 电子科技大学硕士学位论文, 2011 年.
- [6] 田嘉洪. 多机动目标跟踪技术研究. 南京航空航天大学硕士毕业论文, 2007 年.
- [7] 樊宗臣, 吴晓燕. 机动目标跟踪的自适应卡尔曼滤波算法实现. 战术导弹技术, 2009 年 1 月, (1): 79~81.
- [8] 安志忠, 王东进. 多站雷达中机动目标高精度跟踪分析. 系统工程与电子技术, 2004 年 1 月, 第 26 卷第 1 期: 14~17.
- [9] 陈亮. 机动目标跟踪关键技术研究. 哈尔滨工程大学博士学位论文, 2012 年 10 月.
- [10] 陈家军, 王东进, 陈卫东. 基于多基地雷达系统的机动目标跟踪算法. 火控雷达技术, 2006 年 6 月, 第 35 卷: 9~13.
- [11] 尹海滨. 雷达机动目标跟踪滤波算法的研究. 大连海事大学硕士学位论文, 2008 年 6 月.
- [12] 罗笑冰. 强机动目标跟踪技术研究. 国防科学技术大学博士学位论文, 2007 年 9 月.
- [13] 冯洋. 多目标跟踪的数据关联算法研究. 西安电子科技大学硕士学位论文, 2008 年 1 月.
- [14] 王兰云, 赵拥军. 多目标跟踪数据关联及改进算法. 微计算机信息, 2005 年, 第 21 卷第 21-1 期: 190~192.
- [15] 赵艳丽, 林辉, 赵峰, 王雪松, 王国玉. 多目标跟踪中的数据关联和航迹管理. 现代雷达, 2007 年, 第 29 卷第 3 期: 28~31.
- [16] 司锡才, 陈玉坤, 李志刚. 数据关联算法的研究. 2007 年, 第 28 卷第 7 期: 813~817.
- [17] 黄玲. 数据挖掘及融合技术研究与应用. 西北工业大学硕士学位论文, 2004 年 3 月.
- [18] 胡磊力, 陈钰清, 丁贤澄, 邹盛怀. 一种多目标跟踪起始和数据关联的快速算法. 电光与控制, 2000 年, 第 4 期: 5~9.
- [19] 张燕鹏, 薛俊诗, 胡凯. 基于多基地雷达的高超声速目标跟踪研究. 火控雷

-
- 达技术, 2013 年, 第 42 卷第 4 期: 18~21, 27.
- [20] 谭顺成, 王国宏, 王娜, 何友. 基于概率假设密度滤波和数据关联的脉冲多普勒雷达多目标跟踪算法. 电子与信息学报, 2013 年 11 月, 第 35 卷第 11 期: 2700~2706.
- [21] 徐洪奎, 王东进. 基于卡尔曼滤波的组网雷达系统目标跟踪分析. 系统工程与电子技术, 2001 年, 第 23 卷 11 期: 67~69.
- [22] Yaakov Bar-Shalom, X.Rong Li, Thiagalingam Kirubarajan. Estimation with Applications to Tracking and Navigation: Theory Algorithms and Software.
- [23] Bar-Shalom Y, Daum F E, Huang J. The Probabilistic Data Association Filter: Estimation in the Presence of Measurement Origin Uncertainty[J]. IEEE Control Systems Magazine, 2009, 29(6): 82~100
- [24] Sinha A, Ding Z, Kirubarajan T, et al. Track Quality Based Multitarget Tracking Approach for Global Nearest-Neighbor Association[J]. IEEE Transactions on Aerospace and Electronic Systems, 2012, 48(2): 1179~1191
- [25] Fortmann T, Bar-Shalom Y, Scheffe M. Sonar Tracking of Multiple Targets Using Joint Probabilistic Data Association[J]. IEEE Journal of Oceanic Engineering, 1983, 8(3): 173~184.
- [26] X. R. Li and V. P. Jilkov, "A Survey of Maneuvering Target Tracking—Part IV: Decision-Based Methods," In: Proc. SPIE - Signal and Data Processing of Small Targets 2002, April 1–5, 2002, Orlando, FL, USA, pp. 511–534
- [27] X. R. Li and V. P. Jilkov, "Survey of Maneuvering Target Tracking, Part V: Multiple-Model Methods," IEEE Transactions on Aerospace and Electronic Systems, 41(4): 1255–1321, October 2005.
- [28] X. R. Li and V. P. Jilkov, "A Survey of Maneuvering Target Tracking: Approximate Techniques for Nonlinear Filtering," In: Proc. SPIE- Signal and Data Processing of Small Targets 2004, April 13–15, 2004, Orlando, FL, USA, pp. 537–550.
- [29] X. R. Li and V. P. Jilkov, "A Survey of Maneuvering Target Tracking—Part VIa: Density-Based Exact Nonlinear Filtering," Proc. 2010 SPIE Conf. Signal and Data Processing of Small Targets, Orlando, FL, USA, 6–8 April 2010
- [30] X. R. Li and V. P. Jilkov, "A Survey of Maneuvering Target Tracking—Part VIb: Approximate Nonlinear Density Filtering in Mixed Time," Proc. 2010 SPIE Conf. Signal and Data Processing of Small Targets, Orlando, FL, USA, 6–8 April 2010
- [31] Julier S J, Uhlmann J K. A New Extension of The Kalman Filter to Nonlinear Systems[J]. SPIE, 1997, (3068): 182~193
- [32] Julier S J, Uhlmann J K. A New Method for the Nonlinear Transformation of Means and Covariance in Filters and Estimations[J]. IEEE Transactions on Automatic Control, 2000, 45(3): 477~482
- [33] Singer R A. Estimating Optimal Tracking Filter Performance for Manned Maneuvering Targets[J]. IEEE Transactions on Aerospace and Electronic Systems, 1970, AES-6(4): 473~483.
- [34] M. S. Grewal and A. P. Andrew, Kalman Filtering Theory and Practice Using Matlab Third Edition, New York: John Wiley & Sons, pp. 225-289, 2008.

-
- [35]G. Allaire and S.M. Kaber, Numerical Linear Algebra, New York: Springer, 2008.
- [36]R. Hunger, Floating Point Operations in Matrix-Vector Calculus, Technical Report TUM-LNS-TR-05-05. Munich University of Technology, 2005.

参赛体会

写下这段文字已经是 2014 年 9 月 23 日凌晨,距离比赛结束仅有几个小时。虽然我们在参赛论文里撰写参赛体会并不符合一般学术论文的要求,但是历时四天的比赛让我们感慨颇多,既有经验,也有教训,更重要的是体验了一个“分析问题—解决问题”的过程。我们觉得有必要将这一段经历记录下来。

在选题阶段,我们只是简单浏览了一下各题,就选定了 B 题作为我们研究的题目,因为我们认为自己具有一定的航空航天背景,解决这一类题目应该是得心应手。于是,我们在问题研究的初期阶段显得非常浮躁,在建模的过程中考虑的不够深入,导致在解决第一个问题的时候迟迟没有结果。在经历了一次次挫折和打击之后,我们认识到了自己的问题,队伍成员们静下心来,坐在一起站在出题者的角度深入地研究了每一个问题,并最终完成了所有问题的解答。

在整个问题的解答过程中,我们学到了很多以前从未接触过的知识。比如,我们对卡尔曼滤波和卡尔曼预测有了全新的认识,同时,也对算法复杂度的概念有了一定的理解。除此之外,我们在建模过程中提高了自己分析问题、解决问题的逻辑思维能力。回顾整个建模过程,我们收获了许多宝贵经验,也犯下了许多错误,经验教训总结如下:

1) 拿到题目后,一定要审清题目,全方位地考虑问题的解决思路,不能一斑窥豹。

2) 在建模过程中,一旦遇到问题,团队各成员应该坐到一起仔细分析,通过查找参考文献,寻找解决问题的方法。

3) 在建模过程中,团队成员既要相互合作,共同寻找问题解决思路,也要分工明确,否则难以按时完成所有问题的解答。

4) 在建模过程中,一定要一边建模,一边写论文,在写论文的同时,可以发现建模过程中一些未考虑的细节,有助于问题的正确解决。

5) 在建模过程中,遇到棘手的问题,一定要相互鼓励,共同克服困难,只有整个团队充满正能量,才能完成问题的解答。

6) 在建模过程中,一定要合理安排时间,保持精力,否则会在最后冲刺阶段倍感乏力。

历经本次数学建模竞赛,我们学到了很多从未接触过的知识,对我们以后的学术研究有了一定的启发。更重要的是,我们深刻认识到了团队合作的重要性,在以后的科研学习中一定要与身边的人多交流、多学习,共同进步。

附 录

A.问题 1MATLAB 仿真程序

Main.m:

```
clc
clear all
close all
Data1=load('Data1.txt');
Z=Data1;
r=Z(:,1);%距离
psi=Z(:,2)/180*pi;%方位(度)转换为弧度
theta=Z(:,3)/180*pi;%俯仰(度)转换为弧度
t=Z(:,4);
No=Z(:,5);
% % % % 时间处理
for i=2:728
Data1(i,4)=Data1(i,4)-Data1(1,4);
end
Data1(1,4)=0;
XJ=Data1;
for i=1:1:728
XX(i,:)=R2XYZ( r,psi,theta,i,No);
end
save('Data1.mat','Data1');
figure
plot3(XX(:,1),XX(:,2),XX(:,3),'*r')
grid on;
% %
clc;clear all;
% close all;
load Data1.mat
Z=Data1;
r=Z(:,1);%距离
psi=Z(:,2)/180*pi;%方位(度)转换为弧度
theta=Z(:,3)/180*pi;%俯仰(度)转换为弧度
t=Z(:,4);
No=Z(:,5);
clear Data1
T=1;
A=[zeros(3,3) eye(3); zeros(3,6)];
B=[zeros(3,3);eye(3)];
```

```

PHI=expm(A*T);
Tau=[1/2*T^2*eye(3);T*eye(3);];
Xk3=R2XYZ( r,psi,theta,1,No );
Pk=eye(6);
g=9.8;
a_max=1*g;
xigma=0.5*a_max;
Xig=[xigma 0 0; 0 xigma 0; 0 0 xigma];
Q=Tau*Xig*Tau';
n_r=3;
deg2rad=1/180*pi;
R1=[(50/n_r)^2 0 0; 0 (0.4*deg2rad/n_r)^2 0;0 0 (0.4*deg2rad/n_r)^2];
R2=[(40/n_r)^2 0 0; 0 (0.3*deg2rad/n_r)^2 0;0 0 (0.3*deg2rad/n_r)^2];
R3=[(60/n_r)^2 0 0; 0 (0.5*deg2rad/n_r)^2 0;0 0 (0.5*deg2rad/n_r)^2];
XX=[];
TN=749;
Rada1=0;
Rada2=0;
Rada3=0;
for i=1:1:TN
    if i==299
        Xk1=X0f2; %% 赋雷达1最后一点的值为初值
        Xk=Xk1;
        continue;
    end
    find_i=find(t==i);
    sz=size(find_i);
    if isempty(find_i)
        Xk1=PHI*Xk;
        Xk=Xk1;
    else
        if sz(1)==1
            switch No(find_i(1)) %不同雷达数据间隔不同,但是假设相同雷达数据连续采样
            case 1
                if Rada1==0
                    Zk=R2XYZ( r,psi,theta,find_i ,No);
                    Xk_1=[Zk;zeros(3,1)];
                    Xk=Xk_1;
                    Pk_1=eye(6);
                    Rada1=1;
                else
                    Zk=R2XYZ( r,psi,theta,find_i ,No);
                    Xk1_1=PHI*Xk_1;
                    H=[eye(3) zeros(3,3)];

```

```

    Pk1_1=PHI*Pk_1*PHI'+Q;
    J=Jz_R(r( find_i),psi( find_i),theta( find_i));
    RL=J*R1*J';
    K_1=Pk1_1*H'/(H*Pk1_1*H'+RL);
    Pk_1=(eye(6)-K_1*H)*Pk1_1;
    Xk_1=Xk1_1+K_1*(Zk-H*Xk1_1);
    Xk=Xk_1;
end
case 2
if Rada2==0
    Zk=R2XYZ( r,psi,theta,find_i ,No);
    Xk_2=[Zk;zeros(3,1)];
    Pk_2=eye(6);
    Rada2=1;
    Xk=Xk_2;
else
    Zk=R2XYZ( r,psi,theta,find_i ,No);
    Xk1_2=PHI*Xk_2;
    H=[eye(3) zeros(3,3)];
    Pk1_2=PHI*Pk_2*PHI'+Q;
    J=Jz_R(r( find_i),psi( find_i),theta( find_i));
    RL=J*R2*J';
    K_2=Pk1_2*H'/(H*Pk1_2*H'+RL);
    Pk_2=(eye(6)-K_2*H)*Pk1_2;
    Xk_2=Xk1_2+K_2*(Zk-H*Xk1_2);
    Xk=Xk_2;
end
case 3
if Rada3==0
    Zk=R2XYZ( r,psi,theta,find_i ,No);
    Xk_3=[Zk(1:3);zeros(3,1)];
    Pk_3=eye(6);
    Rada3=1;
    Xk=Xk_3;
else
    Zk=R2XYZ( r,psi,theta,find_i ,No);
    Xk1_3=PHI*Xk_3;
    H=[eye(3) zeros(3,3)];
    Pk1_3=PHI*Pk_3*PHI'+Q;
    J=Jz_R(r( find_i),psi( find_i),theta( find_i));
    RL=J*R3*J';
    K_3=Pk1_3*H'/(H*Pk1_3*H'+RL);
    Pk_3=(eye(6)-K_3*H)*Pk1_3;
    Xk_3=Xk1_3+K_3*(Zk-H*Xk1_3);

```

```

        Xk=Xk_3;
    end
end
else
    Zk=R2XYZ( r,psi,theta,find_i(1) ,No);
    Xk1_2=PHI*Xk_2;
    H=[eye(3) zeros(3,3);];
    Pk1_2=PHI*Pk_2*PHI'+Q;
    J=Jz_R(r( find_i(1)),psi( find_i(1)),theta( find_i(1)));
    RL=J*R2*J';
    K_2=Pk1_2*H'/(H*Pk1_2*H'+RL);
    Pk_2=(eye(6)-K_2*H)*Pk1_2;
    Xk_2=Xk1_2+K_2*(Zk-H*Xk1_2);
    if Rada3==0
        Zk=R2XYZ( r,psi,theta,find_i(2) ,No);
        Xk_3=[Zk(1:3);zeros(3,1);];
        Pk_3=eye(6);
        Rada3=1;
    else
        Zk=R2XYZ( r,psi,theta,find_i(2) ,No);
        Xk1_3=PHI*Xk_3;
        H=[eye(3) zeros(3,3);];
        Pk1_3=PHI*Pk_3*PHI'+Q;
        J=Jz_R(r( find_i(2)),psi( find_i(2)),theta( find_i(2)));
        RL=J*R3*J';
        K_3=Pk1_3*H'/(H*Pk1_3*H'+RL);
        Pk_3=(eye(6)-K_3*H)*Pk1_3;
        Xk_3=Xk1_3+K_3*(Zk-H*Xk1_3);
    end
    Xk=0.5*(Xk_2+Xk_3);
end
end
XX=[XX;Xk'];

    if i==254
        X0f2=Xk;
    end

end
grid on;
XX1=XX(1:254,:);
XX2=XX(299:end,:);
XX=[XX1;XX2];
%% 作三维航迹图
hold on

```

```

plot3(XX(:,1), XX(:,2), XX(:,3), 'b','LineWidth',2);
xlabel('x 单位: m')
ylabel('y 单位: m')
zlabel('z 单位: m')
legend('测量数据','估计值')
grid on;
%% 作v-t图
Time=1:1:748;
T1=1:254;
T2=299:748;
vx1=XX1(:,4);
vy1=XX1(:,5);
vx2=XX2(:,4);
vy2=XX2(:,5);
vz1=XX1(:,6);
vz2=XX2(:,6);
figure
subplot 311
plot([T1,T2],[vx1;vx2], 'b','LineWidth',2)
xlabel('t 单位: s')
ylabel('v_{x} 单位: m/s')
subplot 312
plot([T1,T2],[vy1;vy2], 'b','LineWidth',2)
xlabel('t 单位: s')
ylabel('v_{y} 单位: m/s')
subplot 313
plot([T1,T2],[vz1;vz2], 'b','LineWidth',2)
xlabel('t 单位: s')
ylabel('v_{z} 单位: m/s')
%% 作a-t图
temp=diff(vx1);
ax1=[0;temp];
temp=diff(vy1);
ay1=[0;temp];
temp=diff(vz1);
az1=[0;temp];
temp=diff(vx2);
ax2=[0;temp];
temp=diff(vy2);
ay2=[0;temp];
temp=diff(vz2);
az2=[0;temp];
figure
subplot 311

```

```

plot([T1,T2],[ax1;ax2], 'b','LineWidth',2)
xlabel('t 单位: s')
ylabel('a_{x} 单位: m/s^{2}')
subplot 312
plot([T1,T2],[ay1;ay2], 'b','LineWidth',2)
xlabel('t 单位: s')
ylabel('a_{y} 单位: m/s^{2}')
subplot 313
plot([T1,T2],[az1;az2], 'b','LineWidth',2)
xlabel('t 单位: s')
ylabel('a_{z} 单位: m/s^{2}')
%% 坐加速度大小和方向-t图
a_r1=sqrt(ax1.^2+ay1.^2);
a_r2=sqrt(ax2.^2+ay2.^2);
a_phi1=atan2(ax1,ay1);
a_phi2=atan2(ax2,ay2);
a_r1=smooth(a_r1);
a_r2=smooth(a_r2);
a_phi1=smooth(a_phi1);
a_phi2=smooth(a_phi2);
figure
subplot 211
plot([T1,T2],[a_r1;a_r2], 'b','LineWidth',2)
xlabel('t')
ylabel('加速度大小 单位: m/s^{2}')
subplot 212
plot([T1,T2],[a_phi1;a_phi2], 'b','LineWidth',2)
xlabel('t')
ylabel('加速度方向 单位: 弧度')
%%
mean(a_r1(100:160))
mean(a_r2(300-299:320-299))
mean(a_r2(350-299:749-299))

```

R2XYZ.m:

```

function [ out ] = R2XYZ( r,psi,theta,i,No )
p=[r(i)*cos(theta(i))*sin(psi(i)); r(i)*cos(theta(i))*cos(psi(i)); r(i)*sin(theta(i))];
Radius=6400000;
p2=[Radius*0.3/180*pi*cos(40.5*pi/180); Radius*1/180*pi; 0;];
p3=[Radius*0.6/180*pi*cos(40.5*pi/180); Radius*0.4/180*pi; 0;];
switch No(i)
    case 1
        out=p;
    case 2

```

```

        out=p+p2;
    case 3
        out=p+p3;
end

```

Jz_R.m:

```

function [ out ] = Jz_R( r,psi,theta )
Jz=[cos(theta)*sin(psi) r*cos(theta)*cos(psi) -r*sin(theta)*sin(psi);
    cos(theta)*cos(psi) -r*cos(theta)*sin(psi) -r*sin(theta)*cos(psi);
    sin(theta) 0 r*cos(theta)];
out=Jz;
end

```

B.问题 2MATLAB 仿真程序

Main.m:

```

clc;clear all;
close all;
load Data2.mat
Z=Data2;
r=Z(:,1);%距离
psi=Z(:,2);%方位(度)转换为弧度
theta=Z(:,3);%俯仰(度)转换为弧度
t=Z(:,4);
No=Z(:,5);
sz=size(t);
%%
T=0.6;
A=[zeros(3,3) eye(3); zeros(3,6);];
B=[zeros(3,3);eye(3);];
PHI=expm(A*T);
Tau=[1/2*T^2*eye(3);T*eye(3);];
g=9.8;
a_max=1*g;
Xig=0.5*a_max*eye(3);
Q=Tau*Xig*Tau';
n_r=3;
deg2rad=1/180*pi;
R1=[(100/n_r)^2 0 0; 0 (0.3*deg2rad/n_r)^2 0;0 0 (0.3*deg2rad/n_r)^2];
R2=[(100/n_r)^2 0 0; 0 (0.6*deg2rad/n_r)^2 0;0 0 (0.6*deg2rad/n_r)^2];
Xk_1=zeros(6,1);
Xk_2=zeros(6,1);
Pk_1=eye(6);
Pk_2=eye(6);
Total_time=max(t);

```

```

i=0;
time=0;
H=[eye(3) zeros(3,3)];
TN=2649;
M=[];
Y1=[];
Y2=[];
XX1=[];
XX2=[];
%% -----
% 解决思路主要是基于改变最近邻门限值，以及限定目标状态更新间隔两种手段来实现
% 假定不限定更新时间，去掉对Real_T状态更新间隔的判断，则目标1抢走了目标2的所有数据
% 思路一：限定状态更新为0.6秒，则两目标航迹完整；
% 对于目标1不用的数据，更相信它是目标2的数据，因此放大目标2的最近邻门限值
%%
Real_T=0.5;%前后两次状态更新的时间间隔
Target1=struct('fix',0,'t_start',0,'t_current',0,'Gama',50);
Target2=struct('fix',0,'t_start',0,'t_current',0,'Gama',500);
%% %%%%%%%%%%%
% 只有一个回波点迹的情况下，两个目标以目标1优先，数据只属于一个目标。
% 一个数据回波，只能来自同一个源
for i=1:1:TN
    M=Data2(i,:);
    time=M(4);
    Z0=R2XYZ_direct(M(1),M(2),M(3),1);
    position=Z0;
    plot3(position(1), position(2), position(3), 'g*');
    hold on;
    if Target1.fix==0 % 目标1建档
        Target1.fix=1;
        Zm=R2XYZ_direct(M(1,1),M(1,2),M(1,3),1);
        Xk_1=[Zm(1);Zm(2);Zm(3);zeros(3,1)];
        Target1.t_start=M(1,4);
        Target1.t_current=M(1,4);
        M=[];
        position=Xk_1;
        plot3(position(1), position(2), position(3), '*');
        XX1=[XX1;position'];
        hold on;
        Y1=[Y1 [Xk_1;time;]];
    end
    if (Target1.fix==1) && (isempty(M)==0) % 已有目标1的档案
        if (time-Target1.t_current)>Real_T%假设短时间内来了数，则判断属于目标几

```

```

y=M;
Zk=R2XYZ_direct(y(1),y(2),y(3),1);
J=Jz_R(y(1),y(2)*deg2rad,y(3)*deg2rad);
RL1=J*R1*J';
Xk1_1=PHI*Xk_1;
Zk1=Xk1_1(1:3);
Pk1_1=PHI*Pk_1*PHI'+Q;
S1=(H*Pk1_1*H'+RL1);
dk=Zk-Zk1;
d=sqrt(dk'/S1*dk);
if d<Target1.Gama
    Target1.t_current=y(4);
    M=[];
    Xk1_1=PHI*Xk_1;
    H=[eye(3) zeros(3,3)];
    Pk1_1=PHI*Pk_1*PHI'+Q;
    K_1=Pk1_1*H'/(H*Pk1_1*H'+RL1);
    Pk_1=(eye(6)-K_1*H)*Pk1_1;
    Xk_1=Xk1_1+K_1*(Zk-H*Xk1_1);
    position=Xk_1;
    plot3(position(1), position(2), position(3), 'r');
    XX1=[XX1;position'];
    hold on;
    Y1=[Y1 [Xk_1;time;]];
else
end
end
if isempty(M)
else
    if Target2.fix==0 % 目标2建档
        Target2.fix=1;
        Zm=R2XYZ_direct(M(1,1),M(1,2),M(1,3),1);
        Xk_2=[Zm(1);Zm(2);Zm(3);zeros(3,1)];
        Target2.t_start=M(1,4);
        Target2.t_current=M(1,4);
        M=[];
        position=Xk_2;
        plot3(position(1), position(2), position(3), 'r');
        XX2=[XX2;position'];
        hold on;
        Y2=[Y2 [Xk_2;time;]];
    end
    if (Target2.fix==1) && (isempty(M)==0)

```

```

        if (time-Target2.t_current)>Real_T%判断目标2是否需要更新
            y=M;
            Zk=R2XYZ_direct(y(1),y(2),y(3),1);
            J=Jz_R(y(1),y(2)*deg2rad,y(3)*deg2rad);
            RL2=J*R2*J';
            Xk1_2=PHI*Xk_2;
            Zk2=Xk1_2(1:3);
            Pk1_2=PHI*Pk_2*PHI'+Q;
            S2=(H*Pk1_2*H'+RL2);
            dk=Zk-Zk2;
            d=sqrt(dk'/S2*dk);
            if d<Target2.Gama
                Target2.t_current=y(4);
                M=[];
                Xk1_2=PHI*Xk_2;
                H=[eye(3) zeros(3,3)];
                Pk1_2=PHI*Pk_2*PHI'+Q;
                K_2=Pk1_2*H'/(H*Pk1_2*H'+RL2);
                Pk_2=(eye(6)-K_2*H)*Pk1_2;
                Xk_2=Xk1_2+K_2*(Zk-H*Xk1_2);
                position=Xk_2;
                plot3(position(1), position(2), position(3), 'r*');
                XX2=[XX2;position'];
                Y2=[Y2 [Xk_2;time;]];
            else
            end
        end
    end
end

end
figure(2)
plot3(XX1(:,1),XX1(:,2),XX1(:,3),'*b')
hold on
plot3(XX2(:,1),XX2(:,2),XX2(:,3),'*r')
xlabel('x单位: m');
ylabel('y单位: m');
zlabel('z单位: m');
legend('目标一的航迹点','目标二的航迹点');
grid on
%%
%%将原始数据转换成笛卡尔坐标系;
for m=1:1:TN
    XYZ(:,m)=R2XYZ_direct(Data2(m,1),Data2(m,2),Data2(m,3),1);
end

```

```

figure(3)
plot3(XYZ(1,:), XYZ(2,:), XYZ(3,:), 'g*')
hold on
plot3(XX1(:,1),XX1(:,2),XX1(:,3),'.b')
hold on
plot3(XX2(:,1),XX2(:,2),XX2(:,3),'.or')
xlabel('x单位: m');
ylabel('y单位: m');
zlabel('z单位: m');
legend('原始数据','目标一的航迹点','目标二的航迹点');
grid on

```

R2XYZ_direct.m:

```

function [ out ] = R2XYZ_direct( r,psi,theta,i)
psi=psi/180*pi;
theta=theta/180*pi;
p=[r*cos(theta)*sin(psi); r*cos(theta)*cos(psi); r*sin(theta)];
Radius=6400000;
p2=[Radius*cos(40.5/180*pi)*0.3/180*pi; Radius*1/180*pi; 0;];
p3=[Radius*cos(40.5/180*pi)*0.6/180*pi; Radius*0.4/180*pi; 0;];
switch i
    case 1
        out=p;
    case 2
        out=p+p2;
    case 3
        out=p+p3;
end

```

Jz_R.m:

```

function [ out ] = Jz_R( r,psi,theta )
Jz=[cos(theta)*sin(psi) r*cos(theta)*cos(psi) -r*sin(theta)*sin(psi);
    cos(theta)*cos(psi) -r*cos(theta)*sin(psi) -r*sin(theta)*cos(psi);
    sin(theta) 0 r*cos(theta)];
out=Jz;
end

```

C.问题 3MATLAB 仿真程序

Main_1.m:

```

clc;clear all;close all;
load Data3.mat
Z=Data3;
r=Z(:,1);%距离

```

```

psi=Z(:,2);%方位(度)转换为弧度
theta=Z(:,3);%俯仰(度)转换为弧度
t=Z(:,4);
No=Z(:,5);
clear Data3
T=1;
A=[zeros(3,3) eye(3) zeros(3,3);
    zeros(3,6) eye(3);
    zeros(3,9);];
PHI=expm(A*T);
Tau=[1/2*T^2*eye(3);1/2*T*eye(3); eye(3)];
Xk3=R2XYZ_direct3( r(1),psi(1),theta(1));
Pk=eye(9);
g=9.8;
a_max=10*g;%a_max可调节
xigma=0.5*a_max;
Xig=0.5*a_max*eye(3);
Q=Tau*Xig*Tau';
n_r=3;
deg2rad=1/180*pi;
R=[(100/n_r)^2 0 0; 0 (0.5*deg2rad/n_r)^2 0; 0 0 (0.5*deg2rad/n_r)^2];
Xk=[Xk3(1:3);zeros(6,1);];
XJ=[];
TN=528;
Rada1=0;
for i=1:1:TN
    if Rada1==0
        Zk=R2XYZ_direct3(r(i),psi(i),theta(i));
        Xk_1=[Zk;zeros(6,1);];
        Xk=Xk_1;
        Pk_1=eye(9);
        Rada1=1;
    else
        Zk=R2XYZ_direct3(r(i),psi(i),theta(i));
        Xk1_1=PHI*Xk_1;
        H=[eye(3) zeros(3,6);];
        Pk1_1=PHI*Pk_1*PHI'+Q;
        deg2rad=1/180*pi;
        J=Jz_R(r(i),psi(i)*deg2rad,theta(i)*deg2rad);
        RL=J*R*J';
        K_1=Pk1_1*H'/(H*Pk1_1*H'+RL);
        Pk_1=(eye(9)-K_1*H)*Pk1_1;
        Xk_1=Xk1_1+K_1*(Zk-H*Xk1_1);
        Xk=Xk_1;
    end
end

```

```

    end
    XJ=[XJ;Xk'];
end
XX=XJ;
%% 作三维航迹图
figure
plot3(XX(:,1), XX(:,2), XX(:,3), 'b','LineWidth',2);
xlabel('x 单位: m')
ylabel('y 单位: m')
zlabel('z 单位: m')
grid on;
Data3=load('Data3.txt');
% % % 时间处理
for i=2:529
Data3(i,4)=Data3(i,4)-Data3(1,4);
end
Data3(1,4)=0;
XH=Data3;
for i=1:1:528
XXX(i,:)=R2XYZ_direct3( XH(i,1),XH(i,2),XH(i,3));
end
hold on
plot3(XXX(:,1),XXX(:,2),XXX(:,3),'r')
grid on;
legend('估计曲线','测量数据')
%% 作v-t图
Time=0:1:527;
vx=XX(:,4);
vy=XX(:,5);
vz=XX(:,6);
figure
subplot 311
plot(Time,vx, 'b','LineWidth',2)
xlabel('t 单位: s')
ylabel('v_{x} 单位: m/s')
grid on;
subplot 312
plot(Time,vy, 'b','LineWidth',2)
xlabel('t 单位: s')
ylabel('v_{y} 单位: m/s')
grid on;
subplot 313
plot(Time,vz, 'b','LineWidth',2)
xlabel('t 单位: s')

```

```

ylabel('v_{z} 单位: m/s')
grid on;
%% 作 a-t图
ax=XX(:,7);
ay=XX(:,8);
az=XX(:,9);
figure
subplot 311
plot(Time,ax, 'b','LineWidth',2)
xlabel('t 单位: s')
ylabel('a_{x} 单位: m/s^{2}')
grid on;
subplot 312
plot(Time,ay, 'b','LineWidth',2)
xlabel('t 单位: s')
ylabel('a_{y} 单位: m/s^{2}')
grid on;
subplot 313
plot(Time,az, 'b','LineWidth',2)
xlabel('t 单位: s')
ylabel('a_{z} 单位: m/s^{2}')
grid on;

```

Main_2.m:

```

clc;clear all;close all;
load Data3.mat
Z=Data3;
r=Z(:,1);%距离
psi=Z(:,2);%方位(度)转换为弧度
theta=Z(:,3);%俯仰(度)转换为弧度
t=Z(:,4);
No=Z(:,5);
clear Data3
T=1;
A=[zeros(3,3) eye(3); zeros(3,6);];
B=[zeros(3,3);eye(3);];
PHI=expm(A*T);
Tau=[1/2*T^2*eye(3);T*eye(3);];
Xk3=R2XYZ_direct3( r(1),psi(1),theta(1));
Pk=eye(6);
g=9.8;
a_max=g;%a_max可调节
xigma=0.5*a_max;
Xig=0.5*a_max*eye(3);

```

```

Q=Tau*Xig*Tau';
n_r=3;
deg2rad=1/180*pi;
R=[(100/n_r)^2 0 0; 0 (0.5*deg2rad/n_r)^2 0; 0 0 (0.5*deg2rad/n_r)^2];
Xk=[Xk3(1:3);zeros(3,1)];
XJ=[];
TN=528;
Rada1=0;
for i=1:1:TN
    if Rada1==0
        Zk=R2XYZ_direct3(r(i),psi(i),theta(i));
        Xk_1=[Zk;zeros(3,1)];
        Xk=Xk_1;
        Pk_1=eye(6);
        Rada1=1;
    else
        Zk=R2XYZ_direct3(r(i),psi(i),theta(i));
        Xk1_1=PHI*Xk_1;
        H=[eye(3) zeros(3,3)];
        Pk1_1=PHI*Pk_1*PHI'+Q;
        deg2rad=1/180*pi;
        J=Jz_R(r(i),psi(i)*deg2rad,theta(i)*deg2rad);
        RL=J*R*J';
        K_1=Pk1_1*H'/(H*Pk1_1*H'+RL);
        Pk_1=(eye(6)-K_1*H)*Pk1_1;
        Xk_1=Xk1_1+K_1*(Zk-H*Xk1_1);
        Xk=Xk_1;
    end
    XJ=[XJ;Xk'];
end
XX=XJ;
%% 作三维航迹图
figure
plot3(XX(:,1), XX(:,2), XX(:,3), 'b','LineWidth',2);
xlabel('x 单位: m')
ylabel('y 单位: m')
zlabel('z 单位: m')
grid on;
Data3=load('Data3.txt');
%% %% 时间处理
for i=2:529
    Data3(i,4)=Data3(i,4)-Data3(1,4);
end
Data3(1,4)=0;

```

```

XH=Data3;
for i=1:1:528
XXX(i,:)=R2XYZ_direct3( XH(i,1),XH(i,2),XH(i,3));
end
hold on
plot3(XXX(:,1),XXX(:,2),XXX(:,3),'r')
grid on;
legend('估计曲线','测量数据')
%% 作v-t图
Time=0:1:527;
vx=XX(:,4);
vy=XX(:,5);
vz=XX(:,6);
figure
subplot 311
plot(Time,vx, 'b','LineWidth',2)
xlabel('t 单位: s')
ylabel('v_{x} 单位: m/s')
grid on;
subplot 312
plot(Time,vy, 'b','LineWidth',2)
xlabel('t 单位: s')
ylabel('v_{y} 单位: m/s')
grid on;
subplot 313
plot(Time,vz, 'b','LineWidth',2)
xlabel('t 单位: s')
ylabel('v_{z} 单位: m/s')
grid on;
%% 作 a-t图
temp=diff(vx);
ax=[0;temp];
temp=diff(vy);
ay=[0;temp];
temp=diff(vz);
az=[0;temp];
ax=smooth(ax);
ay=smooth(ay);
az=smooth(az);
figure
subplot 311
plot(Time,ax, 'b','LineWidth',2)
xlabel('t 单位: s')
ylabel('a_{x} 单位: m/s^{2}')

```

```

grid on;
subplot 312
plot(Time,ay, 'b','LineWidth',2)
xlabel('t 单位: s')
ylabel('a_{y} 单位: m/s^{2}')
grid on;
subplot 313
plot(Time,az, 'b','LineWidth',2)
xlabel('t 单位: s')
ylabel('a_{z} 单位: m/s^{2}')
grid on;
%% 坐加速度大小和方向-t图
a_r=sqrt(ax.^2+ay.^2+az.^2);
a_phi=atan(ax./ay);
figure
subplot 211
plot(Time,a_r, 'b','LineWidth',2)
xlabel('t')
ylabel('加速度大小 单位: m/s^{2}')
grid on;
subplot 212
plot(Time,a_phi, 'b','LineWidth',2)
xlabel('t')
ylabel('加速度方向 单位: 弧度')
grid on;

```

R2XYZ_direct.m:

```

function [ out ] = R2XYZ_direct( r,psi,theta,i)
psi=psi/180*pi;
theta=theta/180*pi;
p=[r*cos(theta)*sin(psi); r*cos(theta)*cos(psi); r*sin(theta)];
Radius=6400000;
p2=[Radius*cos(40.5/180*pi)*0.3/180*pi; Radius*1/180*pi; 0;];
p3=[Radius*cos(40.5/180*pi)*0.6/180*pi; Radius*0.4/180*pi; 0;];
switch i
    case 1
        out=p;
    case 2
        out=p+p2;
    case 3
        out=p+p3;
end

```

Jz_R.m:

```

function [ out ] = Jz_R( r,psi,theta )
Jz=[cos(theta)*sin(psi) r*cos(theta)*cos(psi) -r*sin(theta)*sin(psi);
    cos(theta)*cos(psi) -r*cos(theta)*sin(psi) -r*sin(theta)*cos(psi);
    sin(theta) 0 r*cos(theta)];
out=Jz;
end

```

D.问题 4MATLAB 仿真程序

Main.m:

```

clc;clear all;close all;
load Data3.mat
Z=Data3;
r=Z(:,1);% 距离
psi=Z(:,2);% 方位(度)转换为弧度
theta=Z(:,3);% 俯仰(度)转换为弧度
t=Z(:,4);
No=Z(:,5);
[m,n]=size(Z);
T=1;
A=[zeros(3,3) eye(3) zeros(3,3);
    zeros(3,6) eye(3);
    zeros(3,9);];
PHI=expm(A*T);
Tau=[1/2*T^2*eye(3);1/2*T*eye(3); eye(3)];
H=[eye(3) zeros(3,6);];
g=9.8;
a_max=10*g;% a_max可调节
xigma=0.5*a_max;
Xig=0.5*a_max*eye(3);
Q=Tau*Xig*Tau';
n_r=3;
deg2rad=1/180*pi;
R=[(100/n_r)^2 0 0; 0 (0.5*deg2rad/n_r)^2 0;0 0 (0.5*deg2rad/n_r)^2];
Xk3=R2XYZ_direct3( r(1),psi(1),theta(1));
Xk=[Xk3(1:3);zeros(6,1);];
%% 一步预测
XP=zeros(9,m);
XP(:,1)=Xk;
Pk=eye(9);
J=Jz_R(r(1),psi(1)*deg2rad,theta(1)*deg2rad);
RL=J*R*J';
for i=2:m
    K=PHI*Pk*H'/(H*Pk*H'+RL);

```

```

    temp=R2XYZ_direct3( r(i-1),psi(i-1),theta(i-1));
    XP(:,i)=PHI*XP(:,i-1)+K*(temp-H*XP(:,i-1));
    Pk=PHI*Pk*PHI'-PHI*Pk*H'/(H*Pk*H'+RL)*H*Pk*PHI'+Q;
    J=Jz_R(r(i),psi(i)*deg2rad,theta(i)*deg2rad);
    RL=J*R*J';
end
load Data4.mat
X=XX;
[m,n]=size(X);
figure
plot3(X(:,1), X(:,2), X(:,3), 'b','LineWidth',2);
hold on
plot3(XP(1,:), XP(2,:), XP(3,:), 'r','LineWidth',2);
grid on;
legend('基于卡尔曼滤波器的目标轨迹估计曲线','基于卡尔曼预测器的目标轨迹预测曲线')
xlabel('x 单位: m')
ylabel('y 单位: m')
zlabel('z 单位: m')
error1=0;
for i=1:m
    error1=error1+sqrt((X(i,1)-XP(1,i))^2+(X(i,2)-XP(2,i))^2+(X(i,3)-XP(3,i))^2);
end
error1=error1/m
%%
Xk_1=XP(:,m);
predict=[];
for i=1:100
    Xk1_1=PHI*Xk_1;
    predict=[predict;Xk1_1'];
    Xk_1=Xk1_1;
    if Xk1_1(3)<=0
        break;
    end
end
figure
plot3(XP(1,:), XP(2,:), XP(3,:), 'b','LineWidth',2);
grid on;
hold on
Data3=load('Data3.txt');
% % % % 时间处理
for i=2:529
    Data3(i,4)=Data3(i,4)-Data3(1,4);
end
Data3(1,4)=0;

```

```

XH=Data3;
for i=1:1:528
XXX(i,:)=R2XYZ_direct3( XH(i,1),XH(i,2),XH(i,3));
end
plot3(XXX(:,1),XXX(:,2),XXX(:,3),'g')
hold on
plot3(predict(:,1), predict(:,2), predict(:,3), 'b','LineWidth',2);
legend('预测曲线','测量数据')
%% 插值求精确落点
XI=-2.245159053792917e+05:1:-1.559684671251006e+05;
YI=5.888211946508928e+04:-0.1:5.115647914293028e+04;
ZI = interp1(predict(:,1), predict(:,3),XI);
Z2 = interp1(predict(:,2), predict(:,3),YI);

```

R2XYZ_direct.m:

```

function [ out ] = R2XYZ_direct( r,psi,theta,i)
psi=psi/180*pi;
theta=theta/180*pi;
p=[r*cos(theta)*sin(psi); r*cos(theta)*cos(psi); r*sin(theta)];
Radius=6400000;
p2=[Radius*cos(40.5/180*pi)*0.3/180*pi; Radius*1/180*pi; 0;];
p3=[Radius*cos(40.5/180*pi)*0.6/180*pi; Radius*0.4/180*pi; 0;];
switch i
    case 1
        out=p;
    case 2
        out=p+p2;
    case 3
        out=p+p3;
end

```

Jz_R.m:

```

function [ out ] = Jz_R( r,psi,theta )
Jz=[cos(theta)*sin(psi) r*cos(theta)*cos(psi) -r*sin(theta)*sin(psi);
    cos(theta)*cos(psi) -r*cos(theta)*sin(psi) -r*sin(theta)*cos(psi);
    sin(theta) 0 r*cos(theta)];
out=Jz;
end

```

computational_complexity_1.m:

```

clc;clear all
l=3;
n=0:10;

```

```

for i=1:11
    y1(i) = 4 * n(i)^3 + 6 * (n(i)^2) * l + 6 * n(i) * l^2 + n(i)^2 - n(i) + l^3;
    y2(i) = 4 * n(i)^3 + 8 * (n(i)^2) * l + 4 * n(i) * l^2 - n(i)^2 - n(i) + l^3;
end
zz1 = 0:10: y1(10);
temp=n(10)*eye(length(zz1));
temp1=diag(temp);
zz2 = 0:10: y2(10);
temp=n(10)*eye(length(zz2));
temp2=diag(temp);
zz3 = 0:0.1: 9;
temp=y1(10)*eye(length(zz3));
temp3=diag(temp);
zz4 = 0:0.1: 9;
temp=y2(10)*eye(length(zz4));
temp4=diag(temp);
plot(n,y1,'g',n,y2,'k','linewidth',4);
xlabel('n:状态维数','FontSize',12,'FontWeight','bold','Color','k');
ylabel('复杂度','FontSize',12,'FontWeight','bold','Color','k');
legend('Kalman估计器','Kalman预测器');
grid on
hold on
plot(temp2,zz2,'-r','linewidth',2.5);
hold on
plot(temp1,zz1,'-b','linewidth',2.5);
hold on
plot(zz4,temp4,'-r','linewidth',2.5);
hold on
plot(zz3,temp3,'-b','linewidth',2.5);
hold on

```

computational_complexity_2.m:

```

clc;clear all
n=9;
l=0:10;
for i=1:11
    y1(i) = 4 * n^3 + 6 * (n^2) * l(i) + 6 * n * l(i)^2 + n^2 - n + l(i)^3;
    y2(i) = 4 * n^3 + 8 * (n^2) * l(i) + 4 * n * l(i)^2 - n^2 - n + l(i)^3;
end
zz1 = 0:10: y1(4);
temp=l(4)*eye(length(zz1));
temp1=diag(temp);
zz2 = 0:10: y2(4);

```

```
temp=l(4)*eye(length(zz2));
temp2=diag(temp);
zz3 = 0 :0.1: 3;
temp=y1(4)*eye(length(zz3));
temp3=diag(temp);
zz4 = 0 :0.1: 3;
temp=y2(4)*eye(length(zz4));
temp4=diag(temp);
plot(l,y1,'g',l,y2,'k','linewidth',4);
xlabel('L:量测维数','FontSize',12,'FontWeight','bold','Color','k');
ylabel('复杂度','FontSize',12,'FontWeight','bold','Color','k');
legend('Kalman估计器','Kalman预测器');
grid on
hold on
plot(temp2,zz2,'-r','linewidth',2.5);
hold on
plot(temp1,zz1,'-b','linewidth',2.5);
hold on
plot(zz4,temp4,'-r','linewidth',2.5);
hold on
plot(zz3,temp3,'-b','linewidth',2.5);
hold on
```